NASA Technical Memorandum 110427

Steady-State and Transient Boundary Element Methods for Coupled Heat Conduction

Dean A. Kontinos

January 1997

National Aeronautics and
Space Administration

NASA Technical Memorandum 110427

Steady-State and Transient Boundary Element Methods for Coupled Heat Conduction

Dean A. Kontinos, Thermosciences Institute, Ames Research Center, Moffett Field, California

January 1997

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

Contents

Nomenclature

Acronyms

BEM	boundary element method
CFD	computational fluid dynamics

Symbols

A, B, C, Q	distance function coefficients
$A_1,\ A_2$	integration coefficients
a, b	nodal offset values
C_p	domain view factor at source point
c	heat capacity coefficient
$c_1,\ c_2$	coefficients
$\mathbf{D}_j$	vector matrix containing domain integration coefficients
e	internal energy per unit volume
e_t	total energy per unit volume
F	arbitrary function
$\mathbf{G}_{jk},\ \mathbf{H}_{jk}$	matrices containing influence coefficients
H	Heaviside function
I_j^{ss}	steady-state component integrals
I_j^{tr}	transient component integrals
K_{jk}	tensor thermal conductivity
k_s	scalar thermal conductivity
l_e	length of linear element
$n,\ \hat{n},\ n_j$	outward unit normal
$\vec{p}$	position vector to the point source
$\mathbf{Q}_j$	vector matrix containing nodal temperature gradients
$\overline{q}$	temperature gradient boundary condition
q_j	heat flux in index notation
$\vec{q}$	position vector to arbitrary point in the domain
r	distance from source point to point in domain
$\vec{r}$	vector from source point to point in domain

S	surface
T	temperature
$\overline{T}$	temperature boundary condition
$\hat{T}$	numerical approximation of the temperature
$\tilde{T}$	numerical approximation of the temperature defined by equations (3.11a) or (4.10a)
T_I	initial temperature
T'	perturbation temperature
$\mathbf{T}_j$	vector matrix containing nodal temperature
T_0	reference temperature
t	time
t_f	final time
t_0	initial time
u	dummy integration variable
u_j, u_k	velocity in index notation
V	volume
W	domain weight function
$\overline{W}$, $\overline{\overline{W}}$	boundary weight functions
W^{ss}	fundamental solution for steady-state conduction
W^{tr}	fundamental solution for transient conduction
x, y	spatial coordinates
x_j	spatial coordinates in index notation
α	thermal diffusivity
Γ	domain boundary
γ	integration limit
δ	Dirac delta function
ε	error residual
η	transformed temporal coordinate
λ, θ, ψ	polar transform variables
ξ	transformed spatial coordinate
ρ	solid density
τ	time

Φ_{jk}	viscous dissipation in index notation
$\tilde{\phi}_e$	solution variable interpolation function
χ_i	local transformed temporal coordinate
Ω	domain
ω	parameter, defined by equation (3.33)

Subscripts

e	denotes element
i, j, k, m	indices
p	denotes source point
q	denotes node point
Γ	denotes boundary
Ω	denotes domain

Superscripts

ss	denotes steady-state
tr	denotes transient

Steady-State and Transient Boundary Element Methods for Coupled Heat Conduction

Dean A. Kontinos

Ames Research Center

Summary

Boundary element algorithms for the solution of steady-state and transient heat conduction are presented. The algorithms are designed for efficient coupling with computational fluid dynamic discretizations and feature piecewise linear elements with offset nodal points. The steady-state algorithm employs the fundamental solution approach; the integration kernels are computed analytically based on linear shape functions, linear elements, and variably offset nodal points. The analytic expressions for both singular and nonsingular integrands are presented. The transient algorithm employs the transient fundamental solution; the temporal integration is performed analytically and the nonsingular spatial integration is performed numerically using Gaussian quadrature. A series solution to the integration is derived for the instance of a singular integrand. The boundary-only character of the algorithm is maintained by integrating the influence coefficients from initial time. Numerical results are compared to analytical solutions to verify the current boundary element algorithms. The steady-state and transient algorithms are numerically shown to be second-order accurate in space and time, respectively.

1 Introduction

Computational science is advancing dramatically in these first decades of the Information Age. Performance gains in computer technology are propelling scientific computing to the forefront of the engineering process. Increasing processor speed is promoting greater fidelity of the physical models, and increases in memory are permitting finer resolution of the physical domain. Simulations, once impossible to perform in a timely fashion, are routinely computed using desktop workstations. Furthermore, computational modeling is extending into every theatre of engineering, for example, fluid dynamics, acoustics, heat transfer, chemistry, astrophysics, and structural mechanics.

Although the maturing numerical methodology is applicable across the engineering spectrum, much of the development is compartmentalized into the separate disciplines. For example, in the aerospace industry, computational fluid dynamics (CFD) simulations performed in the aerodynamics or propulsion group are passed, sometimes blindly, as a load condition to the structures group wherein a structural dynamic simulation is computed. To bridge this gap, the latest effort in computational science is coupling analysis codes across disciplines. This effort is a natural path of development, but, more importantly, it is being driven by engineering systems whose complexity dictates a coupled analysis. For instance, in the silicon chip fabrication industry, accurate simulation of

chemical vapor deposition and chemical etching requires modeling of the fluid dynamics in the reactor coupled to a surface chemistry model. In the aerospace field, advanced hypersonic concepts blend the propulsion system with the body mold line, thereby blurring the traditional industry demarcations of aerodynamics, propulsion, and structure. Also, metallic thermal protection panels are being implemented for the new reusable launch vehicle. Because of aerodynamic heating, the panels expand from the structure and alter the hypersonic flow field. Analysis of the thermal protection system requires coupling of the aerothermodynamics to the structural response. It is in this arena of coupled CFD to structural analysis that the discussion is focused.

Two basic approaches are used to solve coupled fluid/structural systems. The first is a direct coupling where the solution of the different fields is solved simultaneously in one large system of equations. Direct coupling is mostly applicable for problems where time accuracy is critical, such as in aeroelasticity where the time scale of the fluid motion is on the same order as the structural modal frequency. The second approach is a loose coupling strategy where each set of field equations is solved to produce boundary conditions for the other. The equations are solved in turn until an iterated convergence criterion is met at the fluid/solid interface. It is not within the scope of this paper, nor in the expertise of the author, to present a comprehensive literary review of coupled CFD-structural dynamic methods. Instead, reference 1 is recommended, wherein a brief bibliography and a coupled simulation is given.

The loose coupling strategy is particularly attractive when coupling solid mechanics to hypersonic CFD. The time scales of the hypersonic flow field are often disparate from the time scales of the structure, thereby obviating a directly coupled time-accurate analysis. Furthermore, high-speed flows are replete with complex physical phenomena such as shock waves, shock-wave/boundary-layer interactions, chemical reactions, and internal energy exchange. The numerics for hypersonic algorithms are sophisticated and, at times, temperamental. Adding structural equations to the system may diminish the robustness of the crafted hypersonic CFD code. Thus, the loose coupling strategy effectively shields the CFD code from performance degradation while increasing the fidelity of the global simulation.

In the aerospace industry, finite element and finite difference methods are routinely used for the solution of solid heat conduction and elasticity. Well proven, these methods are readily available for loose coupling with a CFD code. A third option, rarely used in the aerospace community, is the boundary element method (BEM). Li and Kassab (refs. 2 and 3) have efficiently solved joint fluid and structural heating by coupling CFD to the BEM solution of the conduction in the body. The advantage of the BEM over a finite difference or finite element formulation is that only the boundary is discretized, and thus the dimensionality of the problem is reduced. As a result, it is naturally coupled with CFD. The boundary element grid is simply the CFD grid at the fluid/surface interface plus additional grid points defining the boundaries of the body. The interior of the domain is not discretized. This reduction in dimensionality is especially advantageous when coupling to CFD because interior values are superfluous; only surface values are required for coupling. Consequently, the BEM is potentially more efficient than finite difference or finite element methods which require an interior discretization to produce surface conditions. Using the BEM, the temperature can be computed at any desired interior point through contour integrals over the boundary solution. In summary, advantages of the BEM are a reduction of dimensionality, ease of discretization, and efficient coupling with CFD.

This paper presents BEM algorithms for two-dimensional, steady-state and transient, heat conduction. The algorithms are specifically designed for efficient coupling with CFD. The presentation includes a brief tutorial on the BEM for those unfamiliar with the technique. Then the details of the current algorithms are presented. The mathematics are overly detailed because the paper is intended as a technical reference manual to document the algorithms. The document is organized as follows: The governing equation of heat conduction is given in Chapter 2, the steady-state algorithm with numerical examples in Chapter 3, and the transient algorithm with numerical examples in Chapter 4.

2 Governing Equations

Consider a material volume in space with no internal heat generation. Applying the conservation of energy, the time rate of change of the volumetric internal energy is equal to the net flux of energy through the bounding surface; in index notation, this relation is

$$\frac{\partial}{\partial t}\int_{Volume} e_t \, dV + \int_{Surface} \left(e_t u_j + \Phi_{jk} u_k + q_j\right) n_j \, dS = 0 \tag{2.1}$$

where e_t is the total energy per unit volume, u_j is the velocity of the particles, Φ_{jk} is the viscous dissipation, q_j is the heat flux, n_j is the outward unit normal, and t is time. In a solid, the material velocity is zero; therefore, the convection and viscous dissipation terms are zero. Furthermore, since the kinetic energy is zero, the total energy comprises only the internal mode. Applying the Divergence Theorem, equation (2.1) becomes

$$\frac{\partial}{\partial t}\int_{Volume} e \, dV + \int_{Volume} \frac{\partial q_j}{\partial x_j} \, dV = 0 \tag{2.2}$$

where e is the internal energy and x_j are the independent spatial variables. Expressions for the internal energy and the heat flux are now established. The internal energy of a solid is empirically determined through the heat capacity, denoted as c, defined as the change in heat content of the solid per change in degree of temperature. In general, c is determined experimentally over a range of temperatures. The relation is

$$c = \frac{1}{\rho}\frac{de}{dT} \tag{2.3}$$

The internal energy is found by integrating cdT from a given reference temperature, T_0, to the temperature of interest,

$$e = \rho \int_{T_0}^{T} c \, dT \tag{2.4}$$

If c is independent of temperature, then the internal energy simplifies to

$$e = \rho c T \tag{2.5}$$

In general, the heat flux comprises radiative and conductive terms. Neglecting radiation, the heat flux is given by Fourier's law, which states that the conductive flux is proportional to the temperature gradient. In index notation, Fourier's law is

$$q_k = -K_{jk} \frac{\partial T}{\partial x_j} \tag{2.6}$$

where K_{jk} is the thermal conductivity. In general, the thermal conductivity is a tensor quantity that is a function of position, direction, and temperature. For this analysis, however, the thermal conductivity is considered to be a scalar quantity denoted as k_s. Thus, modifying equation (2.6) for scalar conductivity and substituting equations (2.5) and (2.6), the energy conservation law (eq. (2.2)), becomes

$$\int_{Volume} \left[\rho c \frac{\partial T}{\partial t} + \frac{\partial}{\partial x_j} \left(-k_s \frac{\partial T}{\partial x_j} \right) \right] dV = 0 \tag{2.7}$$

In equation (2.7), the time derivative is pulled inside the volume integral; this operation is valid as long as the limits of integration are time independent, i.e., the domain is fixed. Letting the volume shrink to zero, the differential form of the heat conduction equation is derived as

$$\rho c \frac{\partial T}{\partial t} = \nabla \bullet \left(k_s \nabla T \right) \tag{2.8}$$

For a constant thermal conductivity equation (2.8) reduces to

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T \tag{2.9}$$

where $\alpha = k_s / (\rho c)$ is the thermal diffusivity. At the steady state, the time derivative of temperature is zero and the heat conduction equation reduces to Laplace's equation, given as

$$\nabla^2 T = 0 \tag{2.10}$$

In summary, the transient heat conduction equation given by equation (2.8) is valid for a solid material with a constant *c*. Further simplification of a constant scalar conductivity yields equation (2.9), which is the transient conduction equation expressed in terms of the thermal diffusivity. Finally, in the steady state, equation (2.9) reduces to Laplace's equation, given by equation (2.10).

3 Steady-State Boundary Element Algorithm

This section develops the boundary element procedure for solving steady-state heat conduction. For completeness, a derivation of the boundary integral equation is presented. The presentation is self-contained, yet is only cursory in detail. Definitive presentations on the BEM are found in references 4 through 6. Regardless of the potential insufficiencies, the derivation of the boundary integral equation is presented through weighted residual analysis. Next, the weight function, which appears in the boundary integral equation as the kernel of an integral transform, is chosen to be Green's free space solution to the governing equation. The free space solution and its directional derivative are specified in this section. Then, shape functions are presented for a linear distribution of the dependent variables over linear boundary elements with offset nodes. Analytic expressions for the integral transforms are given for both singular and nonsingular integrands. Finally, test cases are presented.

3.1 The Boundary Integral Equation

The core of the boundary element method is the boundary integral equation. Equation (2.10) is the starting point for the derivation of the boundary integral equation for steady-state heat conduction. The derivation is presented through the perspective of a weighted residual analysis based upon presentations in references 4 and 6. Let Ω be the solid domain with boundary Γ upon which equation (2.10) is valid. Furthermore, let the boundary be divided into two parts, Γ_1 and Γ_2, for which the following boundary conditions apply:

$$T = \overline{T} \text{ on } \Gamma_1 \tag{3.1a}$$

$$\frac{\partial T}{\partial n} = \overline{q} \text{ on } \Gamma_2 \tag{3.1b}$$

Further subdivision of the boundary does not enhance the validity of the derivation; it only adds to the complexity of the algebra.

The goal of any numerical approximation is to minimize the error in the satisfaction of the governing equation and the boundary conditions. Frequently, the numerical scheme is designed to satisfy the boundary conditions exactly, and the error is minimized on the interior. For this derivation, however, the strict enforcement of the boundary condition is relaxed; the numerical scheme is constructed to minimize the error over the domain and the boundary. Let $\hat{T}$ represent the numerical approximation to *T*, and let ε be the error residual. Over the domain and boundary, the error is given by

$$\varepsilon_\Omega = \nabla^2 \hat{T} \tag{3.2a}$$

$$\varepsilon_{\Gamma_1} = \hat{T} - \overline{T} \tag{3.2b}$$

$$\varepsilon_{\Gamma_2} = \frac{\partial \hat{T}}{\partial n} - \bar{q} \tag{3.2c}$$

In weighted residual analysis, the error residuals are multiplied by weight functions and integrated over the domain and boundary to measure the global error. The weight functions can be viewed as error distribution functions whose choice determines the type of numerical approximation. For example, the method of Galerkin is obtained by choosing the weight functions from the same class of functions used to describe $\hat{T}$. An instructive presentation of the weighted residual approach and its connection to finite difference, finite element, and least-squares techniques is given in reference 6. The weighted residual expression based on the error residuals of equations (3.2) is given as

$$\int_{\Omega} \varepsilon_{\Omega} W \, d\Omega + \int_{\Gamma_1} \varepsilon_{\Gamma_1} \overline{W} \, d\Gamma + \int_{\Gamma_2} \varepsilon_{\Gamma_2} \overline{\overline{W}} \, d\Gamma = 0 \tag{3.3}$$

where W is the weight function over the domain and $\overline{W}$ and $\overline{\overline{W}}$ are the weight functions over the boundary. Substituting in the residual expressions of equations (3.2) yields

$$\int_{\Omega} W \nabla^2 \hat{T} \, d\Omega + \int_{\Gamma_1} \left(\hat{T} - \overline{T} \right) \overline{W} \, d\Gamma + \int_{\Gamma_2} \left(\frac{\partial \hat{T}}{\partial n} - \bar{q} \right) \overline{\overline{W}} \, d\Gamma = 0 \tag{3.4}$$

The boundary integral equation is derived by manipulating the domain integral and judiciously choosing the weight functions. From the weighted residual perspective, the steps of the derivation appear prescient in their introduction; indeed, the source of the foreknowledge is the original formulation from reciprocity considerations. Rizzo (ref. 7) presents an interesting and informative historical view of the boundary integral technique wherein the derivation is far more deductive than that presented herein. Nevertheless, the focus is on transforming the domain integral of equation (3.4) into a more convenient form. This transformation is accomplished by applying Green's identity to the domain integral to reduce the order of the operator on the temperature field. The relation is given by

$$\int_{\Omega} W \nabla^2 \hat{T} \, d\Omega = -\int_{\Omega} \nabla W \bullet \nabla \hat{T} \, d\Omega + \int_{\Gamma} W \nabla \hat{T} \bullet \hat{n} \, d\Gamma \tag{3.5}$$

Substitution of equation (3.5) for the domain integral of equation (3.4) produces the weak form of the residual statement, which is the basis of the finite element method. In the weak form, the weight function is symmetric to the numerical solution and, depending on the choice of the weight function, frequently gives rise to symmetric matrices. To produce the boundary integral equation, Green's identity is applied a second time to transfer the Laplacian operator to the weight function. Transposing the role of $\hat{T}$ and W, equation (3.5) is rearranged to yield

$$\int_{\Omega} \nabla W \bullet \nabla \hat{T} \, d\Omega = -\int_{\Omega} \hat{T} \nabla^2 W \, d\Omega + \int_{\Gamma} \hat{T} \nabla W \bullet \hat{n} \, d\Gamma \tag{3.6}$$

Successive substitution of equations (3.6) and (3.5) into equation (3.4) yields

$$\int_{\Omega} \hat{T}\,\nabla^2 W\,d\Omega - \int_{\Gamma} \hat{T}\,\nabla W \bullet \hat{n}\,d\Gamma + \int_{\Gamma} W\,\nabla \hat{T} \bullet \hat{n}\,d\Gamma + \int_{\Gamma_1} \left(\hat{T} - \overline{T}\right)\overline{\overline{W}}\,d\Gamma + \int_{\Gamma_2} \left(\frac{\partial \hat{T}}{\partial n} - \overline{q}\right)\overline{\overline{W}}\,d\Gamma = 0 \qquad (3.7)$$

By consecutive application of Green's identity, the Laplacian operator in the domain integral has been transferred from the numerical solution to the weight function. This formulation is termed the *inverse problem*, and upon its derivation, attention is turned to the weight functions.

Up to this point, the only limiting assumption is that the weight function, *W*, must be twice differentiable in order to apply Green's theorem consecutively. There is complete freedom in selecting the boundary weights; they are chosen such that

$$\overline{\overline{W}} = \frac{\partial W}{\partial n} \text{ on } \Gamma_1 \qquad (3.8a)$$

$$\overline{\overline{\overline{W}}} = -W \text{ on } \Gamma_2 \qquad (3.8b)$$

Substituting equations (3.8) into equation (3.7) and noting the cancellations in the boundary integrals results in

$$\int_{\Omega} \hat{T}\,\nabla^2 W\,d\Omega - \int_{\Gamma_2} \hat{T}\,\nabla W \bullet \hat{n}\,d\Gamma + \int_{\Gamma_1} W\,\nabla \hat{T} \bullet \hat{n}\,d\Gamma - \int_{\Gamma_1} \overline{T}\,\frac{\partial W}{\partial n}\,d\Gamma + \int_{\Gamma_2} \overline{q} W\,d\Gamma = 0 \qquad (3.9)$$

By consolidating the notation, the boundary integrals can be simplified to yield

$$\int_{\Omega} \tilde{T}\,\nabla^2 W\,d\Omega - \int_{\Gamma} \tilde{T}\,\frac{\partial W}{\partial n}\,d\Gamma + \int_{\Gamma} W \frac{\partial \tilde{T}}{\partial n}\,d\Gamma = 0 \qquad (3.10)$$

where

$$\tilde{T} = \begin{cases} \hat{T} \text{ over } \Omega \text{ and on } \Gamma_2 \\ \overline{T} \text{ on } \Gamma_1 \end{cases} \qquad (3.11a)$$

$$\frac{\partial \tilde{T}}{\partial n} = \begin{cases} \dfrac{\partial \hat{T}}{\partial n} \text{ on } \Gamma_1 \\ \overline{q} \text{ on } \Gamma_2 \end{cases} \qquad (3.11b)$$

It is instructive to review the steps leading to equation (3.10). First, a weighted residual statement is written with separate weight functions for the boundaries and the domain. Green's theorem is applied to yield the inverse problem shown in equation (3.7). Then the boundary weight functions

are selected in terms of the interior weight function to produce cancellation in the boundary integrals. Finally, a judicious variable change simplifies the integral equation.

The remaining task is the selection of the weight function; a profitable choice is Green's free space solution to the governing equation. Green's function is a fundamental solution to the governing equation subject to a unit impulse forcing function. The fundamental solution for steady-state conduction, denoted by W^{ss}, satisfies

$$\nabla^2 W^{ss} = \delta(\vec{q} - \vec{p}) \tag{3.12}$$

where δ is the Dirac delta function, $\vec{q}$ is a position vector to any point in the domain, and $\vec{p}$ is the position vector to the point source. The precise mathematical definition of the Dirac function is ambiguous, but its critical property is that, for a function *F(x)*, the integral of the product of $F(x)\delta(x - x_0)$ satisfies

$$\int_{\Omega} F(x)\ \delta\big(x - x_0\big) d\Omega = F(x_0) \tag{3.13}$$

In some presentations, the Dirac function is defined by equation (3.13), and in some instances, equation (3.13) is a property of the definition; a more extensive discussion of the Dirac function is given in reference 8. In any case, the operation of the Dirac function in lieu of $\nabla^2 W^{ss}$ in the domain integral isolates the value of the temperature at the source point. The domain integral is effectively eliminated. The boundary integral equation becomes

$$C_p \tilde{T}_p - \int_{\Gamma} \tilde{T}\ \frac{\partial W^{ss}}{\partial n}\ d\Gamma + \int_{\Gamma} \mathrm{W}^{ss}\ \frac{\partial \tilde{T}}{\partial n}\ d\Gamma = 0 \tag{3.14}$$

where $\tilde{T}_p$ is the numerical approximation of the temperature at the source point *p*. The coefficient C_p is a function of the included angle exposed to the interior at the source point. Details of the derivation of C_p are given in reference 4.

The boundary integral equation is the core of the boundary element method. By choosing the fundamental solution as the weight function, domain integration has been eliminated; observe from equation (3.14) that only boundary integrals appear. The result of this development is a numerical procedure where the nodal points are located *only* on the boundary. The boundary-only character stands in contrast to finite difference or finite element techniques that require a complete domain discretization; the benefit is a reduction in dimensionality. Furthermore, with the BEM any subset of the interior solution can be calculated to any desired resolution based on the computed boundary solution.

The general outline of the BEM is as follows: The boundary is discretized into elements that can be of any shape, but typically are polynomials as in finite element procedures. The solution variables are assigned an interpolation function based on nodal points distributed over the element. The interpolation function defines the distribution of the solution variable over the element. Originating at the

source point, the fundamental solution and its directional derivative are kernels of an integral transform of the prescribed interpolation functions. After dividing the boundary into elements, the boundary integral equation becomes

$$C_p\tilde{T}_p - \sum_e \int_{\Gamma_e} \tilde{T}_e \frac{\partial W^{ss}}{\partial n}\, d\Gamma + \sum_e \int_{\Gamma_e} \mathrm{W}^{ss} \frac{\partial \tilde{T}_e}{\partial n}\, d\Gamma = 0 \tag{3.15}$$

where *e* denotes an individual element, so $\tilde{T}_e$ and $\partial\tilde{T}_e/\partial n$ denote the distribution of the dependent variables over element *e*. Typically, the integrals are computed numerically using Gaussian quadrature.

Equation (3.15) is written for each node to form a system of linear equations. The system is expressed in matrix notation as

$$\mathbf{H}_{jk}\mathbf{T}_j + \mathbf{G}_{jk}\mathbf{Q}_j = 0 \tag{3.16}$$

where $\mathbf{T}_j$ and $\mathbf{Q}_j$ are vectors containing the nodal temperatures and temperature gradients, respectively, and $\mathbf{H}_{jk}$ and $\mathbf{G}_{jk}$ are matrices containing the influence coefficients resulting from the integral transform. After segregating the known and unknown dependent variables based on the boundary conditions, the linear system is solved to yield the complete solution on the boundary. The algorithms presented in this paper employ Gaussian elimination with partial pivoting for direct inversion of the system matrix. To compute the interior solution, equation (3.15) is applied with the source point located at the interior point of interest. Since the solution on the boundary is completely known, the boundary integrals are computed directly without a matrix inversion.

The remaining ingredients of the numerical recipe are the definition of the fundamental solution and its derivative, the geometrical definition of the boundary element, and the prescription of the approximating interpolation function to the unknown solution variables $\tilde{T}$ and $\partial\tilde{T}/\partial n$. The combination of these ingredients differentiates particular boundary element algorithms. The algorithms presented here are specifically designed for efficient coupling with a CFD flow solver. Serendipitously, an analytic solution of the integral transforms is achieved with the chosen combination of ingredients.

3.2 Fundamental Solution for Two-Dimensional, Steady-State Heat Conduction

The fundamental solution for the two-dimensional Laplace's equation is given by

$$W^{ss} = \frac{1}{2\pi}\ln\frac{1}{r} \tag{3.17}$$

where *r* is the distance from the source point to a point in the domain; it is expressed as

$$r = \|\vec{r}\| = \|\vec{q} - \vec{p}\| \tag{3.18}$$

The directional derivative is given as

$$\frac{\partial W^{ss}}{\partial n} = \frac{1}{2\pi}\frac{\vec{r} \bullet \hat{n}}{r^2} \tag{3.19}$$

Both the fundamental solution and its derivative are singular at $r = 0$, so care must be taken when integrating near or through the source point.

3.3 Element Shape Function

The next ingredient to the numerical recipe is the definition of the boundary element shape. The element shape is distinct from and prescribed independently of the dependent variable distribution. The element shape is a geometrical attribute that determines the distance function, the outward normal, and the integration path of the contour integrals. On the other hand, the dependent variable interpolation function defines the distribution of the dependent variables over the element. The two functions are constrained differently; the element shape is determined by the physical domain, whereas the interpolation function is governed by the variation of the solution over the boundary. The two functions combine to determine the accuracy of the algorithm. The element shape function is given in this section while the variable interpolation function is described in the next.

This boundary element algorithm is specifically designed for coupling with CFD codes that employ finite difference or finite volume techniques. For both structured and unstructured grids, these CFD techniques assume linear segments between grid points. Thus, linear boundary elements are selected in order to ensure one-to-one correspondence of the boundary element grid to the CFD grid. With linear elements, conservation is easily satisfied since interpolation is not required to mate the two domains. Furthermore, employing linear boundary elements creates two simplifications. First, the distance function, *r,* is prescribed analytically between any arbitrary point and line segment. Second, the outward unit normal to a linear element remains constant; consequently, $\vec{r} \bullet \hat{n}$, which arises from the directional derivative of the fundamental solution, is pulled out of the integration. These simplifications allow analytic solution of the integral transforms.

The component expressions for r and $\vec{r} \bullet \hat{n}$ for a linear element are now presented. In two-dimensional space, the source point can be oriented with respect to a linear segment in one of three possible ways. Each orientation results in a different analytic expression and is addressed separately. In *Orientation 1*, the source point is not collinear with the line segment; thus, $\vec{r} \bullet \hat{n} \neq 0$. For *Orientation 2*, the source point and element are collinear but the source point does not lie on the segment itself. Finally, in *Orientation 3*, the source point lies on the element; thus, the element contains an integrable singularity. In both *Orientations 2* and *3*, $\vec{r} \bullet \hat{n} = 0$.

3.3.1 *Orientation 1* Distance Function– Figure 1 displays the notation and schematic of a source point and linear boundary element in *Orientation 1*; $\vec{r}$ is the position vector from the source point to a point on the element; $\vec{r}_{12}$ is the position vector from the first point of the line segment to the end point; and $\vec{r}_{p1}$ is the position vector from the source point to the first point of the line segment. The element is mapped into a linear segment of unit length through the transformation

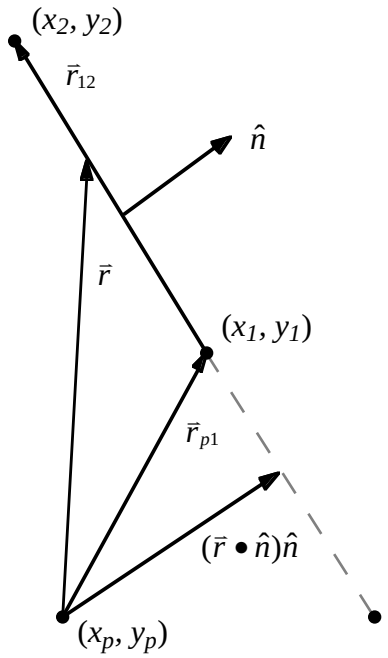

Figure 1. Source point and linear boundary element segment in Orientation 1.

$$x = x_1 + \xi(x_2 - x_1) \tag{3.20a}$$

$$y = y_1 + \xi(y_2 - y_1) \tag{3.20b}$$

where (x_1, y_1) and (x_2, y_2) are the start and end coordinates, respectively, of the line segment defining the element, and ξ is the transform variable with range $0 \le \xi \le 1$. The square of the distance function varies as a quadratic function of the transform variable according to

$$r^2 = \vec{r} \bullet \vec{r} = A + B\xi + C\xi^2 \tag{3.21}$$

where

$$A = \vec{r}_{p1} \bullet \vec{r}_{p1} \tag{3.22a}$$

$$B = 2\vec{r}_{p1} \bullet \vec{r}_{12} \tag{3.22b}$$

$$C = \vec{r}_{12} \bullet \vec{r}_{12} \tag{3.22c}$$

The coefficients *A*, *B*, and *C* are solely functions of the geometry; they attain different values for each element and source point combination.

The magnitude of the dot product of the position vector and the element outward unit normal can be expressed in terms of the distance function coefficients as

$$|\vec{r} \bullet \hat{n}| = \frac{\sqrt{Q}}{2\sqrt{C}} \tag{3.23}$$

where

$$Q = 4AC - B^2 \tag{3.24}$$

By ordering the elements in a counterclockwise fashion by convention, the sign of the dot product is determined by

$$\text{sign}(\vec{r} \bullet \hat{n}) = \text{sign}(\vec{r}_{p1} \times \vec{r}_{12}) \tag{3.25}$$

Note that the dot product becomes zero when Q is zero.

3.3.2 ***Orientation 2*** **Distance Function–** In *Orientation 2*, the source point is collinear with the element. As will be shown later, the analytic expressions of the integration become undefined when the source point and line segment are collinear, i.e., when $Q = 0$. This degeneracy is avoided by redefining the distance function. Since the source point and element are collinear, the location of the source point can be expressed in terms of the element transform variable ξ. Let ξ_p be the position of the source point. Recall that the element is defined over the range $0 \le \xi \le 1$; thus, if $0 \le \xi_p \le 1$, then the source point lies on the line segment resulting in *Orientation 3*. In *Orientation 2*, the source lies off the element, so $\xi_p > 1$ or $\xi_p < 0$. In either case, ξ_p is determined as

$$\xi_p = \frac{-B}{2C} \tag{3.26}$$

The distance function becomes

$$r = \begin{cases} \sqrt{A} - \xi\sqrt{C} & \text{for } \xi_p > 1 \\ \sqrt{A} + \xi\sqrt{C} & \text{for } \xi_p < 0 \end{cases} \tag{3.27}$$

3.3.3 ***Orientation 3*** **Distance Function–** In the third orientation, the source point lies on the element and hence the integral transform is singular at a point along the path of integration. Nevertheless, the integral transform exists in the Cauchy principle value sense. In order to compute the principle value, it is convenient to set the origin of the transform variable at the source point. Such a transformation is dependent on the location of the source point with respect to the grid points; consequently, it is dependent on the solution variable shape function. Therefore, the discussion of the distance function for *Orientation 3* is delayed until after the presentation of the solution variable interpolation function in the next section.

3.4 Solution Variable Interpolation Function

The final ingredient of the numerical recipe is the dependent variable interpolation function. Barring an algorithm designed for a specific application, it is common to set the interpolation function to the same order as the element shape because, in general, the least accurate function will constrain the global accuracy. Thus, the distribution of the solution variables is selected to be linear. Two node points are required on each element to properly define a linear distribution. Normally in finite difference and finite element methods the node points are coincident with the grid points or, in other words, the node points are located at the ends of the linear element. For piecewise linear segments, however, the element normal direction will most likely change discontinuously from one element to the next. Consequently, when the nodal points are coincident with the grid points, the node normal direction, which is used to define $\partial T/\partial n$, becomes nonunique at the node. There are procedures to account for this effect, which amount to substituting an auxiliary equation when relating the upstream and downstream temperature gradients of a particular grid-node point (for example, see ref. 9). In the present work, the node points are variably offset from the ends of the line segment. The placement of the node away from the grid point uniquely defines the element normal at the node. The drawback of the offsetting strategy is an increase in the number of nodes since, in general, two unique nodes per element are required, whereas a node located at the grid point is common to the two adjoining elements. This offsetting procedure is not required in all circumstances. For instance, if two adjoining elements are collinear, then the node point is placed at the grid point without concern. Since minimizing the number of nodes to reduce the computational effort is desirable, a shape function that accounts for variable positioning of the node is described.

Figure 2 shows an element and the position of the nodal points. The nodes are offset from the beginning and end grid points by transformed distances *a* and *b*, respectively. If the value of *a* or *b* is zero, then the grid and node point are coincident. Let $\tilde{\phi}_e$ represent either of the solution variables $\tilde{T}$ or $\partial\tilde{T}/\partial n$ over element *e*. Also, let $\tilde{\phi}_{e,1}$ and $\tilde{\phi}_{e,2}$ be the values of the solution variable at the beginning and end nodes of element *e*, respectively. The solution variable is given by the linear function

$$\tilde{\phi}_e = \frac{1}{(1-b-a)}\left[\tilde{\phi}_{e,1}(1-b-\xi) + \tilde{\phi}_{e,2}(\xi-a)\right] \tag{3.28}$$

Although not indicated by a subscript *e*, the values of *a* and *b* vary from element to element, depending on the local geometry and boundary conditions.

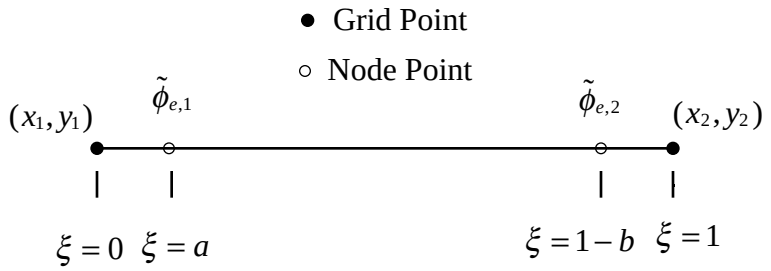

Figure 2. Linear boundary element definition.

3.5 Integral Transforms

Definitions of the weight function, the element shape, and the variable distribution function are now complete. These elements are combined to derive analytic expressions for the integral transforms. Two integrals are computed for each element and source point combination. The integrals are mapped into the transform variable ξ and are given as

$$\int_{\Gamma_e} \tilde{T}_e \frac{\partial W^{ss}}{\partial n} d\Gamma = l_e \int_0^1 \tilde{T}_e \frac{\partial W^{ss}}{\partial n} d\xi \tag{3.29a}$$

$$\int_{\Gamma_e} W^{ss} \frac{\partial \tilde{T}_e}{\partial n} d\Gamma = l_e \int_0^1 W^{ss} \frac{\partial \tilde{T}_e}{\partial n} d\xi \tag{3.29b}$$

where l_e is the length of the element. Recall that the distance function and dot product contained in the fundamental solution and the solution variables have all been previously expressed in terms of ξ. The integral expressions are analytically derived for each of the three possible orientations.

3.5.1 ***Orientation 1*** **Integrals–** In this orientation, the fundamental solution, with the expression for the distance function given by equation (3.21), is combined with the shape function of equation (3.28) to produce the integral transforms. The integrals of equations (3.29) are of similar form

$$l_e \int_0^1 \tilde{T}_e \frac{\partial W^{ss}}{\partial n} d\xi = \frac{\sqrt{Q}}{4\pi(1-b-a)} \left\{ \tilde{T}_{e,1} \left[(1-b) I_1^{ss} - I_2^{ss} \right] + \tilde{T}_{e,2} \left[I_2^{ss} - a I_1^{ss} \right] \right\} \tag{3.30a}$$

$$l_e \int_0^1 W^{ss} \frac{\partial \tilde{T}_e}{\partial n} d\xi = \frac{\sqrt{C}}{4\pi(1-b-a)} \left\{ \left(\frac{\partial \tilde{T}}{\partial n} \right)_{e,1} \left[(1-b) I_3^{ss} - I_4^{ss} \right] + \left(\frac{\partial \tilde{T}}{\partial n} \right)_{e,2} \left[I_4^{ss} - a I_3^{ss} \right] \right\} \tag{3.30b}$$

The terms I_j^{ss} denote component integrals resulting from algebraic manipulation of the integrand. The superscript *ss* indicates steady-state integral components and is used to distinguish those integrals from similar transient integrals presented in the next chapter. The component integrals are derived analytically as

$$I_1^{ss} = \int_0^1 \frac{d\xi}{r^2} = \frac{2}{\sqrt{Q}} \left[\tan^{-1}\left(\frac{2C+B}{\sqrt{Q}} \right) - \tan^{-1}\left(\frac{B}{\sqrt{Q}} \right) \right] \tag{3.31a}$$

$$I_2^{ss} = \int_0^1 \frac{\xi d\xi}{r^2} = \frac{1}{2C} \left(\ln|A+B+C| - \ln|A| - B I_1^{ss} \right) \tag{3.31b}$$

$$I_3^{ss} = \int_0^1 \ln\left|r^2\right| d\xi = \left(1 + \frac{B}{2C} \right) \ln|A+B+C| - \frac{B}{2C} \ln|A| + \frac{Q}{2C} I_1^{ss} - 2 \tag{3.31c}$$

$$I_4^{ss} = \int_0^1 \xi \ln\left|r^2\right| d\xi = \frac{1}{2}\ln|A+B+C| - \frac{B}{2} I_5^{ss} - C I_6^{ss} \tag{3.31d}$$

$$I_5^{ss} = \int_0^1 \frac{\xi^2 d\xi}{r^2} = \frac{1}{C} - \frac{B}{2C^2}\left(\ln|A+B+C| - \ln|A|\right) + \frac{B^2 - 2AC}{2C^2} I_1^{ss} \tag{3.31e}$$

$$I_6^{ss} = \int_0^1 \frac{\xi^3 d\xi}{r^2} = \frac{1}{C}\left(\frac{1}{2} - B I_5^{ss} - A I_2^{ss}\right) \tag{3.31f}$$

3.5.2 ***Orientation 2*** **Integrals–** In the second orientation, $\vec{r} \bullet \hat{n} = 0$ and therefore the integral of equation (3.29a) is zero. The expression for equation (3.29b) is given by equation (3.30b) with the integrals I_3^{ss} and I_4^{ss} derived from the distance function given by equation (3.27). The integrals are given as

$$I_3^{ss} = 2\left[\left(1 + \frac{\sqrt{A}}{\omega}\right)\ln\left|\sqrt{A} + \omega\right| - 1 - \frac{\sqrt{A}}{\omega}\ln\left|\sqrt{A}\right|\right] \tag{3.32a}$$

$$I_4^{ss} = \left(1 - \frac{A}{C}\right)\ln\left|\sqrt{A} + \omega\right| + \frac{\sqrt{A}}{\omega} - \frac{1}{2} + \frac{A}{C}\ln\left|\sqrt{A}\right| \tag{3.32b}$$

where

$$\omega = \begin{cases} \sqrt{C} \text{ for } \xi_p < 0 \\ -\sqrt{C} \text{ for } \xi_p > 1 \end{cases} \tag{3.33}$$

3.5.3 ***Orientation 3*** **Integrals–** As in *Orientation 2*, $\vec{r} \bullet \hat{n} = 0$; therefore, the integral of equation (3.29a) is zero. In the third orientation, the integrand of equation (3.29b) is singular at the source point, but the integral exists in the Cauchy principle value sense. The derivation is accomplished by recasting the distance and interpolation functions in terms of a coordinate system originating at the source point. It is possible for the source point to be located at either of the two nodes on the element, and the singular integration is similar for both possibilities. A slight change in notation condenses the equations of the two possibilities into a single expression. Let subscript *p* denote the value of the gradient at the source node and let subscript *q* denote the other node. In this notation, *a* and *b* are the distances separating points *p* and *q* from the grid point, respectively. This notation departs from the previous definition, which associated *a* and *b* with the ordering of the element nodes. In this new definition, the ordering is arbitrary and the equations are applicable for either node point as source. The integral is given as

$$\int_{\Gamma_e} W^{ss} \frac{\partial \tilde{T}_e}{\partial n} d\Gamma = \frac{l_e}{2\pi}\left[\frac{\partial \tilde{T}_p}{\partial n}(A_1 - A_2) + \frac{\partial \tilde{T}_q}{\partial n} A_2\right] \tag{3.34}$$

where

$$A_1 = \ln|l_e| - 1 + (1-a)\ln|1-a| + a\ln|a| \tag{3.35a}$$

$$A_2 = \frac{1}{2(1-b-a)}\left[(1-2a)\left(\ln|l_e| - \frac{1}{2}\right) + (1-a)^2 \ln|1-a| + a^2 \ln|a|\right] \tag{3.35b}$$

When $a = 0$, the previous expressions for the coefficients A_1 and A_2 are undefined because of the natural log function; however, the limiting form is found to be

$$A_1 = \ln|l_e| - 1 \tag{3.36a}$$

$$A_2 = \frac{1}{2(1-b)}\left[\ln|l_e| - \frac{1}{2}\right] \tag{3.36b}$$

The numerical recipe is complete. Numerical solutions are generated by looping over all node points and elements to determine the orientation. Then the appropriate analytic solutions are used to compute the elements of the global system matrix. Subsequent inversion of the system matrix produces the boundary solution.

3.6 Numerical Results

The previously developed steady-state algorithm is tested by comparing the numerical solution to three separate analytic solutions. The first test case is a cylindrical geometry with imposed surface temperatures. The second test case is a pin-cushion-like geometry formed by the intersection of four circular arcs. The third test case is a seven-sided nonconvex polygon.

3.6.1 Cylinder– The first test problem is the computation of the temperature distribution in a cylinder with an inner radius of 1 unit and an outer radius of 2 units, as given by Becker (ref. 5). The temperatures on the inner and outer radii are 10 and 6, respectively. The exact solution is given by

$$T = -5.771\ \ln r + 10 \tag{3.37}$$

The boundary is discretized using 24 elements on the quarter plane: 8 on each of the inner and outer radii and 4 on each of the remaining two sides. The temperature is prescribed on the inner and outer radii and symmetry boundary conditions ($\partial\tilde{T}/\partial n = 0$) are imposed on the two sides. The steady-state temperature contours of the boundary element solution are compared to the analytic solution in figure 3. The contours demonstrate the radial symmetry of the numerical solution both on the interior and along the symmetry boundary. Furthermore, it is seen that the BEM solution compares very well to the exact solution. The jaggedness of the inner and outer radii curves on the boundary element side of the plot is a result of the piecewise linear segments used to describe the circle.

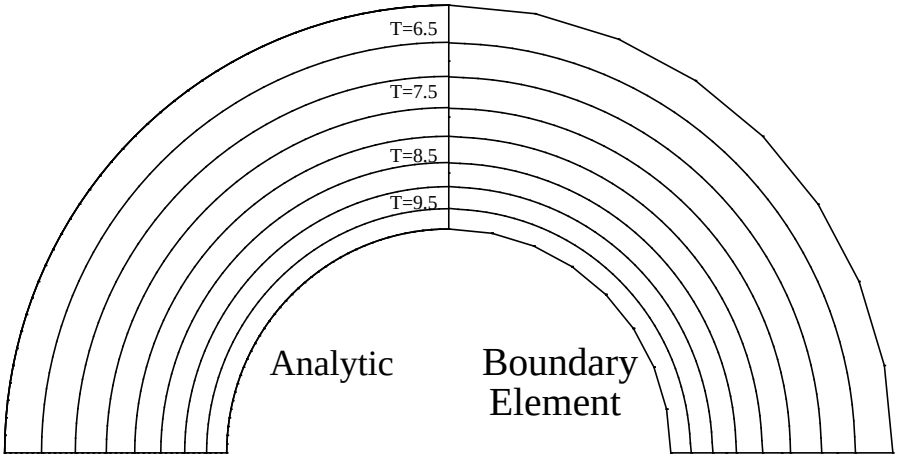

Figure 3. Comparison of analytic and boundary element steady-state temperature contours in a cylinder.

3.6.2 Pin Cushion– The second test case is found in reference 9; it is a pin-cushion geometry defined by the intersection of four circular arcs centered at $(\pm 1, \pm 3)$ with a radius of 3.64, as shown in figure 4. The geometry contains four corners located at the intersection of the circular arcs. Potential deficiencies of the offset node strategy are most likely to appear at such surface contour discontinuities. The exact solution is taken to be

$$T(x,y) = x^2 - y^2 + \ln\left|\left[(x-2)^2 + (y-2)^2\right]^{-\frac{1}{2}}\right| \tag{3.38}$$

Contour lines of the exact solution are shown in figure 4. It should be noted that the temperature field is not symmetric about any axis, consequently, the entire domain must be computed.

The boundary is discretized using 8 elements of equal length on each side for a total of 32 elements. The analytic temperature is imposed on the boundary. The computed surface temperature gradient is compared to the analytic solution in table 1. Comparisons are made upstream and downstream (moving counterclockwise) at the four corners of the pin cushion; typically, the errors in the computed gradients are greatest at the corners. The BEM gradient is within 5 percent of the analytic flux at the selected points despite the potential inaccuracies generated by offset nodes. A similar comparison by Kassab and Nordlund (ref. 9) shows their method to produce errors less than 0.1 percent. The improved accuracy of their method results from the use of quadratic elements, which not only model the surface properties to higher order accuracy but are also able to reproduce the exact surface geometry; the linear elements used in this study generate error from approximating the circular arc shape as a series of line segments.

For the solution reported in table 1, the nodes are offset from the end points by 10 percent of the element length ($a = b = 0.1$). The solution is fairly independent of the offset distance in the range

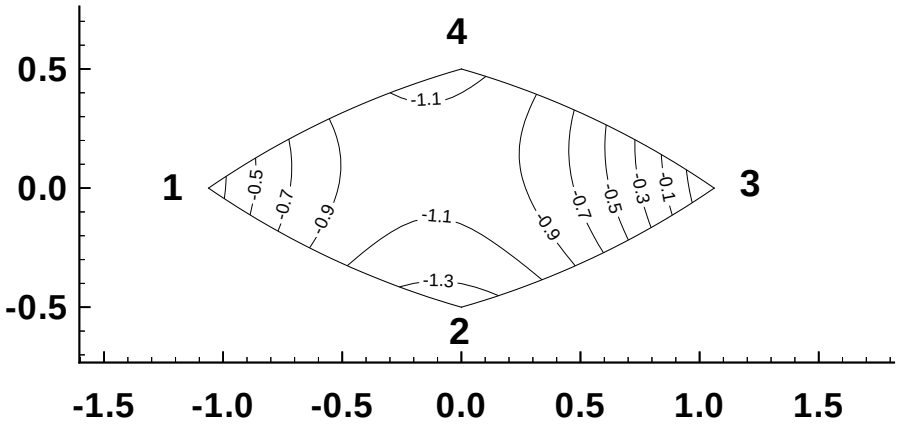

Figure 4. Pin-cushion geometry and analytic temperature contours.

Table 1. Comparison of analytic and BEM temperature gradient at the corners of the pin cushion test problem

Point	Analytic $\partial T/\partial n$	BEM $\partial T/\partial n$	|Error|	Percent error
Corner 1 upstream	1.1959	1.1758	0.0201	–1.68
Corner 1 downstream	0.9494	0.9094	0.0400	–4.21
Corner 2 upstream	–1.2495	–1.2489	0.0006	–0.05
Corner 2 downstream	–1.1423	–1.1280	0.0143	–1.25
Corner 3 upstream	0.9735	0.9501	0.0234	–2.36
Corner 3 downstream	1.6489	1.5978	0.0511	–3.10
Corner 4 upstream	–0.6427	–0.6370	0.0057	–0.89
Corner 4 downstream	–0.8186	–0.8151	0.0035	–0.43

$0.01 < a,b < 0.25$. In general, experience has shown that $a = b = 0.1$ is the most reliable offset value; all the results reported herein are computed using a 10-percent nodal offset. Also, the integral transforms are computed using 16-point Gaussian quadrature to verify the exact integration equations outlined previously. The gradient values generated using Gaussian quadrature prove to be within

4 significant digits of the exact integration results. Additionally, the interior temperature field is computed based on the surface solution. Temperature contours compare within the plotting accuracy of figure 4.

The spatial accuracy of the algorithm is measured by computing an error norm for successive increases in the number of elements. The L_2 error norm is shown in figure 5 as a function of the number of boundary elements for both the boundary gradient and the interior temperature. The error norms of figure 5 are normalized by the respective values of the error using 16 elements. The results show the calculation of the boundary gradient to be first-order accurate while the computation of the interior temperature is second-order accurate. This result is consistent with the accuracy of linear elements employed in the finite element method.

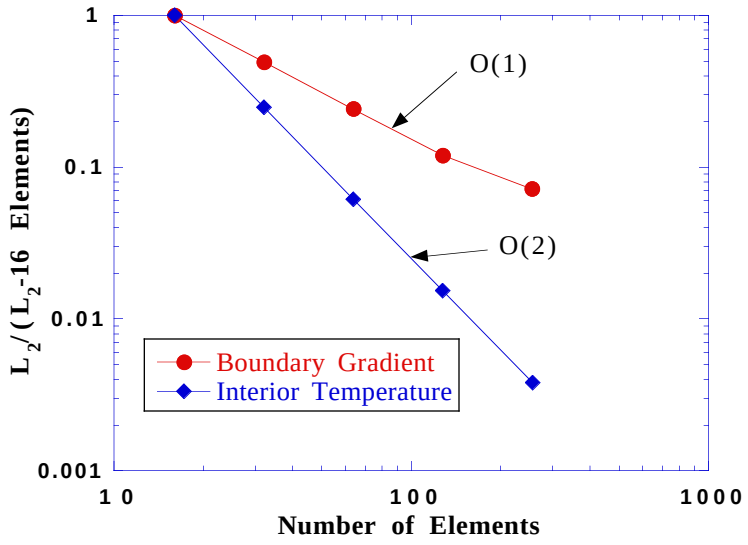

Figure 5. Computed error of pin-cushion solution.

3.6.3 Nonconvex Polygon– The final case is a nonconvex polygon, also extracted from reference 9. The geometry, which is shown in figure 6, is designed to test a range of corner angles. Also shown in figure 6 are contours of the exact solution, given by

$$T(x,y) = \sin(x)\cosh(y) \tag{3.39}$$

The boundary is discretized using 26 elements matching that of reference 9. The analytic temperature is imposed on the boundary. A comparison of the computed boundary temperature gradient is given in figure 7. The abscissa of the plot is surface length starting at $(x,y) = (1,1)$ and proceeding counterclockwise around the domain. Also displayed are the grid point locations. The discontinuities in the gradient values result from the discontinuous change in the surface normal between sides of the polygon. As seen in figure 7, the computed temperature gradient is within 5 percent of the

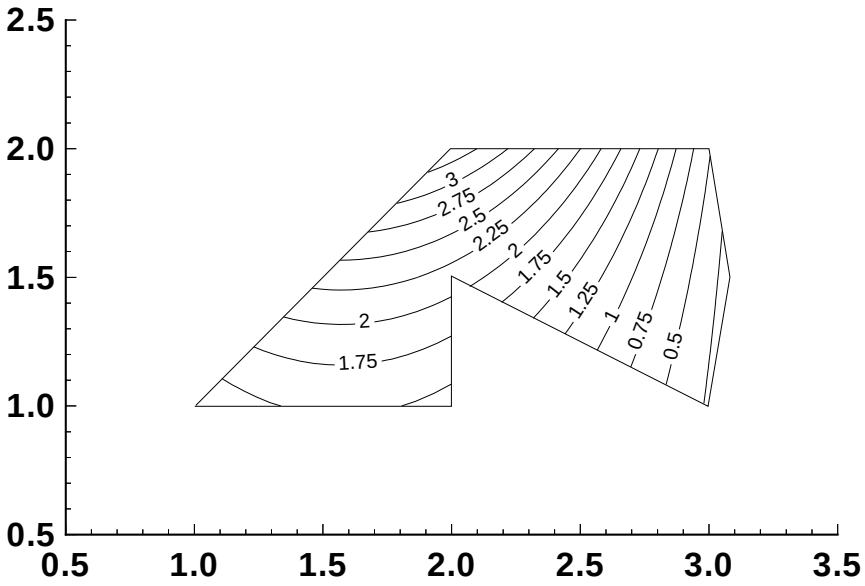

Figure 6. Nonconvex polygon geometry and analytic temperature contours.

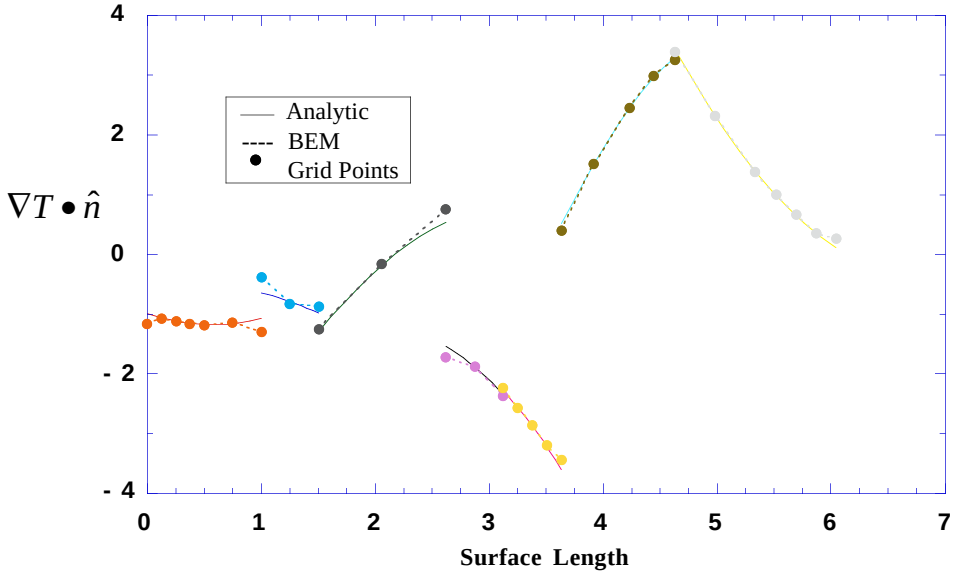

Figure 7. Comparison of analytic and computed temperature gradient for nonconvex polygon.

analytic value on the boundary except at the corner points. The worst comparison is at the 90° corner at $(x,y)=(2,1)$. It is unclear why the maximum error occurs at this location. The solution is not particularly ill-behaved at the corner, suggesting that the effect is geometry governed. It is believed that this result is particular to the offset node strategy. In reference 9, the error at the corners correlated to the turn angle with the 117° corner at $(x,y)=(2,1.5)$ showing the greatest error; no such correlation is found in the solution shown in figure 7. In any case, the efficacy of the solution procedure is demonstrated. Finally, although not shown but worthy of mention, contours of the interior temperature field compare to the analytic solution within the plotting accuracy shown in figure 6.

4 Transient Boundary Element Algorithm

The development of the boundary element method for transient heat conduction follows the same path as the steady-state presentation given in Chapter 3. In principle, the procedure is the same as the steady-state, however, the mathematics are ubiquitous, and the general procedure is easily obscured by the mathematical details. Referral to Chapter 3 may illuminate Chapter 4.

4.1 The Boundary Integral Equation

The derivation of the boundary integral equation commences with the transient heat conduction equation expressed in terms of the thermal diffusivity given by equation (2.9) and repeated as follows:

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0 \text{ in } \Omega \tag{4.1}$$

with time-varying boundary conditions given by

$$T = \bar{T}(t) \text{ on } \Gamma_1 \tag{4.2a}$$

$$\alpha \frac{\partial T}{\partial n} = \bar{q}(t) \text{ on } \Gamma_2 \tag{4.2b}$$

The transient temperature field is split into an initial temperature field, T_I, and a perturbation temperature, T'. The relationship expressed in terms of functional dependence is

$$T(x_j,t) = T'(x_j,t) + T_I(x_j) \tag{4.3}$$

Furthermore, it is assumed that the transient calculation is computed from a steady-state initial condition, i.e.; $\nabla^2 T_I = 0$. Accounting for these relationships, equation (4.1) becomes

$$\frac{\partial T'}{\partial t} - \alpha \nabla^2 T' = 0 \text{ in } \Omega \tag{4.4}$$

Equation (4.4) is the basis for the development of the boundary integral equation. Since the governing equation is written in terms of the perturbation temperature, boundary and initial conditions must be expressed in like terms. The initial conditions are $T'(x_j, t=0) = 0$ by definition. The boundary conditions are recast in terms of the perturbation temperature as

$$T' = \overline{T}(t) - T_I \text{ on } \Gamma_1 \tag{4.5a}$$

$$\alpha \frac{\partial T'}{\partial n} = \overline{q}(t) - \alpha \frac{\partial T_I}{\partial n} \text{ on } \Gamma_2 \tag{4.5b}$$

Next, the domain and boundary error residuals are constructed:

$$\varepsilon_\Omega = \frac{\partial \hat{T}}{\partial t} - \alpha \nabla^2 \hat{T} \tag{4.6a}$$

$$\varepsilon_{\Gamma_1} = \hat{T} - \overline{T} + T_I \tag{4.6b}$$

$$\varepsilon_{\Gamma_2} = \alpha \frac{\partial \hat{T}}{\partial n} - \overline{q} + \alpha \frac{\partial T_I}{\partial n} \tag{4.6c}$$

where $\hat{T}$ is the numerical approximation to T'. Integrating over the domain, boundaries, and time, the weighted residual statement becomes

$$\int_\tau \int_\Omega \left(\frac{\partial \hat{T}}{\partial t} - \alpha \nabla^2 \hat{T} \right) W \, d\Omega \, d\tau + \int_\tau \int_{\Gamma_1} \left(\hat{T} - \overline{T} + T_I \right) \overline{W} \, d\Gamma \, d\tau + \int_\tau \int_{\Gamma_2} \left(\alpha \frac{\partial \hat{T}}{\partial n} - \overline{q} + \alpha \frac{\partial T_I}{\partial n} \right) \overline{\overline{W}} \, d\Gamma \, d\tau = 0 \tag{4.7}$$

where W is the interior weight function, $\overline{W}$ and $\overline{\overline{W}}$ are the weight functions on the boundaries, and τ represents the time domain. Up to this point, the procedure is analogous to the steady-state derivation with the additional dimension of time. The inverse problem is derived by using Green's theorem to transfer the Laplacian operator from the temperature field to the weight function. In addition, the integral of the time derivative of temperature is transformed through the relation

$$\int_\tau \int_\Omega \frac{\partial \hat{T}}{\partial t} W \, d\Omega \, d\tau = \int_\Omega \left(\hat{T} W \Big|_{t_f} - \hat{T} W \Big|_{t_0} - \int_{t_0}^{t_f} \hat{T} \frac{\partial W}{\partial t} \, d\tau \right) d\Omega \tag{4.8}$$

where t_0 and t_f are the initial and final time, respectively. The boundary weight functions are selected in terms of the interior weight function as

$$\overline{W} = -\alpha \frac{\partial W}{\partial n} \text{ on } \Gamma_1 \tag{4.9a}$$

$$\overline{\overline{W}} = W \text{ on } \Gamma_2 \tag{4.9b}$$

Furthermore, the notation is consolidated through the variable change given by

$$\tilde{T} = \begin{cases} \hat{T} \text{ over } \Omega \text{ and on } \Gamma_2 \\ \overline{T} - T_I \text{ on } \Gamma_1 \end{cases} \tag{4.10a}$$

$$\alpha \frac{\partial \tilde{T}}{\partial n} = \begin{cases} \alpha \dfrac{\partial \hat{T}}{\partial n} \text{ on } \Gamma_1 \\ \overline{q} - \alpha \dfrac{\partial T_I}{\partial n} \text{ on } \Gamma_2 \end{cases} \tag{4.10b}$$

After the application of Green's theorem, the substitutions of equations (4.8 through 4.10), and some algebra, the weighted residual statement becomes

$$\begin{aligned} \int_\Omega \left(\tilde{T}W\Big|_{t_f} - \tilde{T}W\Big|_{t_0} \right) d\Omega - \int_\tau \int_\Omega \tilde{T} \left(\frac{\partial W}{\partial t} + \alpha \nabla^2 W \right) d\Omega \, d\tau \\ -\alpha \int_\tau \int_\Gamma W \frac{\partial \tilde{T}}{\partial n} \, d\Gamma \, d\tau + \alpha \int_\tau \int_\Gamma \tilde{T} \frac{\partial W}{\partial n} \, d\Gamma \, d\tau = 0 \end{aligned} \tag{4.11}$$

In order to remove the second domain integral, the weight function for the transient equation, denoted as W^{tr}, is selected to satisfy the following:

$$\frac{\partial W^{tr}}{\partial t} + \alpha \nabla^2 W^{tr} = -\delta(r)\delta(t) \tag{4.12}$$

The right-hand side parameters of equation (4.12) are Dirac functions, which represent a unit impulse forcing function in space and time. In addition, Wrobel and Brebbia (ref. 10) have shown that

$$\int_\Omega \tilde{T} W^{tr} \Big|_{t_f} \, d\Omega = 0 \tag{4.13}$$

Applying equations (4.12 and 4.13) to equation (4.11) yields the transient boundary integral equation, given by

$$-\int_\Omega \tilde{T} W^{tr} \Big|_{t_0} d\Omega + C_p \tilde{T}_p - \alpha \int_\tau \int_\Gamma W^{tr} \frac{\partial \tilde{T}}{\partial n} d\Gamma \, d\tau + \alpha \int_\tau \int_\Gamma \tilde{T} \frac{\partial W^{tr}}{\partial n} d\Gamma \, d\tau = 0 \tag{4.14}$$

Other than the domain integral at $t = t_0$, the transient boundary integral equation is identical in form to the steady-state boundary integral equation (3.14). The differences are the weight function, the additional dimension of time, and the presence of the domain integral, which appears to prevent

boundary-only discretization. Nevertheless, the numerical methodology is identical to the steady-state algorithm. Moreover, it is shown in Section 4.3 that the calculation of the domain integral can be circumvented to recover the boundary-only character of the algorithm. Before that analysis, the transient fundamental solution is introduced.

4.2 Fundamental Solution for Two-Dimensional, Transient Heat Conduction

The fundamental solution satisfying equation (4.12) is given by

$$W^{tr} = \frac{1}{4\pi\alpha(t_f - t)} \exp\left(\frac{-r^2}{4\alpha(t_f - t)} \right) H(t_f - t) \tag{4.15}$$

where $H(t_f - t)$ is the Heaviside function. The fundamental solution is singular when $r = 0$ and $t = t_f$. The directional derivative is given by

$$\frac{\partial W^{tr}}{\partial n} = \frac{-(\vec{r} \bullet \hat{n})}{8\pi\alpha^2 (t_f - t)^2} \exp\left(\frac{-r^2}{4\alpha(t_f - t)} \right) H(t_f - t) \tag{4.16}$$

Both the fundamental solution and its directional derivative are functions of r and $(t_f - t)$; i.e., $W^{tr} = W^{tr}(r, t_f - t)$. This functional dependence is important in the development of the time-step procedure presented in the next section.

When computing the transient integrals, it is convenient to transform the fundamental solution to normalized coordinates. The spatial coordinate is mapped to the parameter ξ previously introduced in Chapter 3. Recall that the distance function, r, is expressed in terms of ξ. Time is transformed into the coordinate η, which originates at the source time level, t_f, and is scaled by the time interval $(t_f - t_0)$. The transformation is backward in time and is given by

$$\eta = \frac{t_f - t}{t_f - t_0} \tag{4.17}$$

The transform variable η ranges from 0 at $t = t_f$ to 1 at $t = t_0$. The fundamental solution and its derivative are written in terms of the transformed variables as

$$W^{tr}(\xi, \eta, t_f - t_0) = \frac{1}{4\pi\alpha(t_f - t_0)\eta} \exp\left(\frac{-r^2(\xi)}{4\alpha(t_f - t_0)\eta} \right) H(\eta(t_f - t_0)) \tag{4.18}$$

$$\frac{\partial W^{tr}(\xi, \eta, t_f - t_0)}{\partial n} = \frac{-(\vec{r} \bullet \hat{n})}{8\pi\alpha^2 (t_f - t_0)^2 \eta^2} \exp\left(\frac{-r^2(\xi)}{4\alpha(t_f - t_0)\eta} \right) H(\eta(t_f - t_0)) \tag{4.19}$$

The fundamental solution and its derivative are functionally dependent on ξ, η, and the time interval $(t_f - t_0)$ from the temporal coordinate transformation.

4.3 Time-Step Procedure

Consider the boundary integral equation expressed over the temporal domain from t_0 to t_f, given as

$$-\int_{\Omega} \tilde{T} W^{tr}(r, t_f - t)\Big|_{t_0} d\Omega + C_p \tilde{T}_p - \alpha \int_{t_0}^{t_f} \int_{\Gamma} W^{tr}(r, t_f - t) \frac{\partial \tilde{T}}{\partial n} d\Gamma \, d\tau + \alpha \int_{t_0}^{t_f} \int_{\Gamma} \tilde{T} \frac{\partial W^{tr}(r, t_f - t)}{\partial n} d\Gamma \, d\tau = 0 \tag{4.20}$$

The functional dependency of the fundamental solution is included in the previous expression to accentuate the source point, which is always located at t_f. After the boundary is discretized, the integral equation can be written in terms of coefficient matrices of the temperature and gradient vectors. The form of the equation is dependent on the order of the temporal accuracy. For discussion purposes, the temporal accuracy is assumed to be O(1) in order to simplify the equations. In practice, the algorithms presented in this paper employ a linear interpolation function, the details of which are presented in Sections 4.4 and 4.5. As before in the steady-state analysis, the notation is switched to matrix form for compactness. Equation (4.20) becomes

$$\mathbf{D}_j(\mathbf{T}_j(t_0), t_f - t_0) + \mathbf{H}_{jk}(t_f - t_0)\mathbf{T}_j(t_f) + \mathbf{G}_{jk}(t_f - t_0)\mathbf{Q}_j(t_f) = 0 \tag{4.21}$$

where $\mathbf{H}_{jk}$ and $\mathbf{G}_{jk}$ are the coefficient matrices, identical in function to those of equation (3.16), and $\mathbf{D}_j$ is a vector containing the domain integral, which is a function of the temperature field at t_0 and the time interval $(t_f - t_0)$. The parenthetical expressions after $\mathbf{T}_j$ and $\mathbf{Q}_j$ indicate time level, while those after $\mathbf{D}_j$, $\mathbf{H}_{jk}$, and $\mathbf{G}_{jk}$ denote functional dependency. As will be shown later, the temporal portion of the integral transforms admit analytic solutions that are functions of the time interval $(t_f - t_0)$. Thus, the coefficient matrices $\mathbf{D}_j$, $\mathbf{H}_{jk}$, and $\mathbf{G}_{jk}$ are denoted as functionally dependent on the time interval. Although not explicitly indicated, the matrices are also functions of the geometry.

Wrobel and Brebbia (ref. 10) present two methods for advancing the boundary integral equation in time. In the first approach, the time integration is initiated from the previous time step; this process requires domain integration and, consequently, a domain grid. In the second approach, domain integration is avoided by initiating the integration for every time step from t_0. The two approaches compel different storage and computation strategies. Both methods are now discussed.

Let the time domain be divided into intervals with the time at the end of each interval denoted by t_m. In the first approach, the solution from t_{m-1} to t_m is given by

$$\mathbf{D}_j(\mathbf{T}_j(t_{m-1}), t_m - t_{m-1}) + \mathbf{H}_{jk}(t_m - t_{m-1})\mathbf{T}_j(t_m) + \mathbf{G}_{jk}(t_m - t_{m-1})\mathbf{Q}_j(t_m) = 0 \qquad (4.22)$$

Equation (4.22) is essentially a rewrite of equation (4.21) with a change in the time interval from $(t_f - t_0)$ to $(t_m - t_{m-1})$. To advance the solution in time, the coefficients in $\mathbf{H}_{jk}$ and $\mathbf{G}_{jk}$ are computed based on the local time step, $(t_m - t_{m-1})$. The domain integral is computed based on the temperature field at the previous time step. For varying values of the time step, $\mathbf{H}_{jk}$ and $\mathbf{G}_{jk}$ need not be retained. If the time step is constant, however, the coefficient matrices are a function of a fixed geometry and a fixed time step; therefore, they can be computed once and stored. This procedure is very efficient except for the requisite domain integration.

In the second method, domain integration is avoided by writing equation (4.21) from t_0 to t_m. At the initial time, $\mathbf{T}_j(t_0) = 0$ by definition (recall that T_j contains the perturbation temperature and not the absolute temperature, which can be nonzero), consequently, the domain integral is zero. Equation (4.20) becomes

$$\sum_{i=1}^{m} \mathbf{H}_{jk}(t_m - t_{i-1})\mathbf{T}_j(t_i) + \sum_{i=1}^{m} \mathbf{G}_{jk}(t_m - t_{i-1})\mathbf{Q}_j(t_i) = 0 \qquad (4.23)$$

The summation appearing in equation (4.23) results from dividing the time integral into intervals corresponding to the time-step distribution. The integrals are computed using the transient history of the dependent variables. The coefficient matrices must be calculated for every combination of $(t_m - t_i)$ for every time step. For example, assume the solution has been computed up to time t_1. To advance the solution to time t_2, coefficient matrices must computed for time intervals of $(t_2 - t_0)$ and $(t_2 - t_1)$. The terms of the first summation of equation (4.23) are

$$\sum_{i=1}^{2} \mathbf{H}_{jk}(t_2 - t_{i-1})\mathbf{T}_j(t_i) = \mathbf{H}_{jk}(t_2 - t_0)\mathbf{T}_j(t_1) + \mathbf{H}_{jk}(t_2 - t_1)\mathbf{T}_j(t_2) \qquad (4.24)$$

At the next time step, the matrices must be computed for time intervals of $(t_3 - t_0)$, $(t_3 - t_1)$, and $(t_3 - t_2)$. The terms of the first summation are

$$\sum_{i=1}^{3} \mathbf{H}_{jk}(t_3 - t_{i-1})\mathbf{T}_j(t_i) = \mathbf{H}_{jk}(t_3 - t_0)\mathbf{T}_j(t_1) + \mathbf{H}_{jk}(t_3 - t_1)\mathbf{T}_j(t_2) + \mathbf{H}_{jk}(t_3 - t_2)\mathbf{T}_j(t_3) \qquad (4.25)$$

Thus, for a calculation of $m_{\max}$ nonuniform time steps, $m_{\max}(m_{\max} + 1)/2$ coefficient matrices must be computed to cover the possible combinations of $(t_m - t_i)$ for $0 < i < m$ and $1 \le m \le m_{\max}$. In general, the value of $(t_m - t_i)$ need not repeat itself, therefore, $\mathbf{H}_{jk}(t_m - t_i)$ and $\mathbf{G}_{jk}(t_m - t_i)$ are never repeated. Storage is minimal because only one matrix at a time is needed, but the amount of computation is high.

Matters are simplified somewhat for a constant time step because the number of matrix calculations is reduced to $m_{\max}$. Furthermore, the matrices, which are a function of integer multiples of the time step, are computed once and stored. For a constant time step, equation (4.23) becomes

$$\sum_{i=1}^{m} \mathbf{H}_{jk}((m-i+1)\Delta t)\mathbf{T}_j(t_i) + \sum_{i=1}^{m} \mathbf{G}_{jk}((m-i+1)\Delta t)\mathbf{Q}_j(t_i) = 0 \tag{4.26}$$

Unfortunately, because the origin of the source point is always at t_m, the matrix vector multiplications of $\mathbf{H}_{jk}((m-i+1)\Delta t)\mathbf{T}_j(t_i)$ and $\mathbf{G}_{jk}((m-i+1)\Delta t)\mathbf{Q}_j(t_i)$ are never repeated. For example, at the second time step, $\mathbf{H}_{jk}(2\Delta t)$ is multiplied by $\mathbf{T}_j(t_1)$, while at the next time step it is multiplied by $\mathbf{T}_j(t_2)$. Consequently, there is no possibility for computational and memory savings by storing intermediate vector-matrix multiplication results.

Wrobel and Brebbia (ref. 11) recommend the first method, which requires domain discretization, for most applications. Although the first method has the overhead of the domain integral, the efficiency is independent of the number of time steps. In the second method, on the other hand, the efficiency decreases as the number of time steps increases; the efficiency of the second method quickly falls below that of the first as the number of time steps is increased. Yet, it is the author's opinion that the primary benefit of using the BEM is the boundary-only discretization. If the domain must be discretized, perhaps the finite element method is better suited since it is conceptually less complicated and is amendable to increases in fidelity such as modeling nonlinearity. In light of this philosophical perspective, the transient algorithm presented in this paper employs the second method to exploit the advantage of boundary-only discretization. Of course, these brush strokes are broad, and certainly there may be compelling reasons to use a BEM that requires domain discretization. It has to be the engineer's discretion when to use a wrench and when to use a ratchet.

4.4 Element Shape Functions and Integral Transforms

This section presents the numerical details of the transient boundary element algorithm. The transient algorithm retains the features developed for the steady state: a linear boundary element shape for efficient coupling with CFD, variably offset nodes for unique definition of the node normal, and a linear spatial distribution of the dependent variables. These components are incorporated into the boundary-only time-step procedure introduced in the previous section. The interpolation function is modified to linearly approximate the dependent variables over space and time. The interpolation function is inserted into the boundary integral equation to derive the integral expressions. The integral transforms over time and space do not admit a complete analytic solution except in the instance of a singularity. The integral transforms of the singular and nonsingular types are addressed separately.

The boundary integral equation for the boundary-only time-step procedure is given as

$$C_p\tilde{T}_p - \alpha\int_{t_0}^{t_m}\int_{\Gamma} W^{tr}(r,t_m-t)\frac{\partial\tilde{T}}{\partial n}d\Gamma\, d\tau + \alpha\int_{t_0}^{t_m}\int_{\Gamma}\tilde{T}\frac{\partial W^{tr}}{\partial n}(r,t_m-t)d\Gamma\, d\tau = 0 \tag{4.27}$$

where t_m is the time at which the solution is to be computed. It is assumed that the solution has been previously computed up to time t_{m-1}. As outlined in the previous section, the domain integral vanishes because the integration always commences from t_0. The time intervals are not necessarily constant; that assumption will be made later in the derivation. The boundary is divided into elements

and the time integration is divided according to the time-step distribution to yield the double summation in the boundary integral equation,

$$C_p\tilde{T}_p - \alpha\sum_{i=1}^{m}\int_{t_{i-1}}^{t_i}\sum_{e}\int_{\Gamma_e} W^{tr}(r,t_m-t)\frac{\partial\tilde{T}_e^i}{\partial n}d\Gamma\, d\tau + \alpha\sum_{i=1}^{m}\int_{t_{i-1}}^{t_i}\sum_{e}\int_{\Gamma_e}\tilde{T}_e^i\frac{\partial W^{tr}}{\partial n}(r,t_m-t)d\Gamma\, d\tau = 0 \qquad (4.28)$$

where $\tilde{T}_e^i$ is the distribution of $\tilde{T}$ over element *e* from time t_{i-1} to t_i, and $\partial T_e^i/\partial n$ is defined likewise. Isolating an individual term from each of the double summations reveals the two integral transforms that comprise the crux of the numerical solution,

$$\int_{t_{i-1}}^{t_i}\int_{\Gamma_e} T_e^i\frac{\partial W^{tr}(r,t_m-t)}{\partial n}d\Gamma\, d\tau \qquad (4.29a)$$

$$\int_{t_{i-1}}^{t_i}\int_{\Gamma_e} W^{tr}(r,t_m-t)\frac{\partial T_e^i}{\partial n}d\Gamma\, d\tau \qquad (4.29b)$$

The solution of the integrals is different for singular and nonsingular integrands. In the nonsingular case, a mixed analytical-numerical solution is prescribed. In the singular case, an analytic series solution is derived.

4.4.1 Nonsingular Integrals– In this section, the equations for the integral transforms of equations (4.29) are presented for the case of a nonsingular integrand. As will be shown, the temporal integration admits an analytic solution, whereas the spatial integration is performed numerically. First, the interpolation function of the dependent variables is developed. Let $\tilde{\phi}_e^i$ represent the distribution of either $\tilde{T}$ or $d\tilde{T}/dn$ over element *e* from time t_{i-1} to t_i. The four-point basis function is given as

$$\begin{aligned}\tilde{\phi}_e^i = \frac{1}{(1-b-a)}\Big\{&\chi_i\Big[(1-b-\xi)\tilde{\phi}_{e,1}(t_{i-1}) + (\xi-a)\tilde{\phi}_{e,2}(t_{i-1})\Big] \\ &+ (1-\chi_i)\Big[(1-b-\xi)\tilde{\phi}_{e,1}(t_i) + (\xi-a)\tilde{\phi}_{e,2}(t_i)\Big]\Big\}\end{aligned} \qquad (4.30)$$

where χ_i is a local time variable that ranges from 1 when $t = t_{i-1}$ to 0 when $t = t_i$. The parenthetical expressions after $\tilde{\phi}_{e,j}$ indicate time level. The subscripts 1 and 2, first introduced in the steady-state interpolation function, indicate the two nodes of each element. Equation (4.30) is a bilinear function over space and time expressed in terms of the local time variable, χ_i, and the spatial transform variable ξ. The interpolation function is more suitably expressed in terms of the global time variable η given by equation (4.17), which in this usage ranges from 1 at $t = t_0$ to 0 at $t = t_m$. Equation (4.30) becomes

$$\tilde{\phi}_e^i = \frac{1}{(1-b-a)}\frac{1}{(t_i - t_{i-1})}\Big\{ -(t_m - t_i)\Big[(1-b)\tilde{\phi}_{e,1}(t_{i-1}) - a\tilde{\phi}_{e,2}(t_{i-1})\Big]$$
$$+ (t_m - t_{i-1})\Big[(1-b)\tilde{\phi}_{e,1}(t_i) - a\tilde{\phi}_{e,2}(t_i)\Big]$$
$$+ (t_m - t_0)\Big[(1-b)\tilde{\phi}_{e,1}(t_{i-1}) - a\tilde{\phi}_{e,2}(t_{i-1}) - (1-b)\tilde{\phi}_{e,1}(t_i) + a\tilde{\phi}_{e,2}(t_i)\Big]\eta$$
$$+ \Big[(t_m - t_i)\Big(\tilde{\phi}_{e,1}(t_{i-1}) - \tilde{\phi}_{e,2}(t_{i-1})\Big) - (t_m - t_{i-1})\Big(\tilde{\phi}_{e,1}(t_i) - \tilde{\phi}_{e,2}(t_i)\Big)\Big]\xi$$
$$+ (t_m - t_0)\Big[-\tilde{\phi}_{e,1}(t_{i-1}) + \tilde{\phi}_{e,2}(t_{i-1}) + \tilde{\phi}_{e,1}(t_i) - \tilde{\phi}_{e,2}(t_i)\Big]\xi\eta \Big\} \quad (4.31)$$

Since the interpolation functions are expressed in terms of ξ and η, they are combined with the transformed fundamental solution given by equations (4.18) and (4.19). The integrals of equations (4.29) are mapped into (ξ-η) space to yield

$$\int_{t_{i-1}}^{t_i}\int_{\Gamma_e} \tilde{T}_e^i \frac{\partial W^{tr}(r, t_m - t)}{\partial n} d\Gamma d\tau = \frac{l_e}{(t_m - t_0)}\int_{\eta(t_i)}^{\eta(t_{i-1})}\int_0^1 \tilde{T}_e^i(\xi,\eta)\frac{\partial W^{tr}\big(r(\xi),\eta,t_m - t_0\big)}{\partial n} d\xi\, d\eta \quad (4.32a)$$

$$\int_{t_{i-1}}^{t_i}\int_{\Gamma_e} W^{tr}(r, t_m - t)\frac{\partial \tilde{T}_e^i}{\partial n} d\Gamma d\tau = \frac{l_e}{(t_m - t_0)}\int_{\eta(t_i)}^{\eta(t_{i-1})}\int_0^1 W^{tr}\big(r(\xi),\eta,t_m - t_0\big)\frac{\partial \tilde{T}_e^i(\xi,\eta)}{\partial n} d\xi\, d\eta \quad (4.32b)$$

The integrals are functions of ξ, η, the known boundary solution for $t \le t_{i-1}$, the boundary conditions, and the unknown quantities at $t = t_i$. The expression does not admit analytic solution over space and time, but manipulation of the temporal integral is fruitful.

The multiplication of the fundamental solution (eqs. (4.18) and (4.19)) and the interpolation function (eq. (4.31)) produces three core time integrals of the form

$$\int_{\eta(t_i)}^{\eta(t_{i-1})} \frac{1}{\eta^k}\exp\left(\frac{-r^2(\xi)}{4\alpha\eta(t_m - t_0)}\right)d\eta \quad \text{for } k = 0, 1, 2 \quad (4.33)$$

The values of η at the limits of integration are found by applying its definition given by equation (4.17). Furthermore, the time integral is split into two parts to yield

$$\int_{\eta(t_i)}^{\eta(t_{i-1})} \frac{1}{\eta^k}\exp\left(\frac{-r^2(\xi)}{4\alpha\eta(t_m - t_0)}\right)d\eta = \int_0^{\frac{t_m - t_{i-1}}{t_m - t_0}} \frac{1}{\eta^k}\exp\left(\frac{-r^2(\xi)}{4\alpha\eta(t_m - t_0)}\right)$$
$$- \int_0^{\frac{t_m - t_i}{t_m - t_0}} \frac{1}{\eta^k}\exp\left(\frac{-r^2(\xi)}{4\alpha\eta(t_m - t_0)}\right) \quad \text{for } k = 0, 1, 2 \quad (4.34)$$

Both of the integrals on the right-hand side of equation (4.34) are of the same form, and a general solution appears to be possible. Yet, the time difference in the exponential function is $(t_m - t_0)$ while the time difference in the numerator of the upper limit of integration is $(t_m - t_i)$; this fact suggests that the integral must be computed for all combinations of $(t_m - t_i)$ and $(t_m - t_0)$ for $0 < i < m$ and $1 \le m \le m_{\max}$. Fortunately, the integrals can be transformed to yield standardized integration limits of $0 \le \eta \le 1$. The results for each of the three types of integrals are

$$\int_0^{\frac{t_m - t_i}{t_m - t_0}} \exp\left(\frac{-r^2}{4\alpha(t_m - t_0)\eta}\right) d\eta = \frac{t_m - t_i}{t_m - t_0} \int_0^1 \exp\left(\frac{-r^2}{4\alpha(t_m - t_i)\eta}\right) d\eta \tag{4.35a}$$

$$\int_0^{\frac{t_m - t_i}{t_m - t_0}} \frac{1}{\eta} \exp\left(\frac{-r^2}{4\alpha(t_m - t_0)\eta}\right) d\eta = \int_0^1 \frac{1}{\eta} \exp\left(\frac{-r^2}{4\alpha(t_m - t_i)\eta}\right) d\eta \tag{4.35b}$$

$$\int_0^{\frac{t_m - t_i}{t_m - t_0}} \frac{1}{\eta^2} \exp\left(\frac{-r^2}{4\alpha(t_m - t_0)\eta}\right) d\eta = \frac{t_m - t_0}{t_m - t_i} \int_0^1 \frac{1}{\eta^2} \exp\left(\frac{-r^2}{4\alpha(t_m - t_i)\eta}\right) d\eta \tag{4.35c}$$

Notice that the integration limits and the time difference in the exponential function have been standardized. Combined with the spatial integration, six component integrals result. The integrals are of standard form, and for a constant time step, are computed once, stored, and used repeatedly. The six integrals are denoted by the superscript *tr* to distinguish them from the steady-state integrals; they are given as

$$I_{1,i}^{tr} = \int_0^1 \int_0^1 \exp\left(\frac{-r_i^2(\xi)}{\eta}\right) d\eta \, d\xi \tag{4.36a}$$

$$I_{2,i}^{tr} = \int_0^1 \int_0^1 \xi \exp\left(\frac{-r_i^2(\xi)}{\eta}\right) d\eta \, d\xi \tag{4.36b}$$

$$I_{3,i}^{tr} = \int_0^1 \int_0^1 \frac{1}{\eta} \exp\left(\frac{-r_i^2(\xi)}{\eta}\right) d\eta \, d\xi \tag{4.36c}$$

$$I_{4,i}^{tr} = \int_0^1 \int_0^1 \frac{\xi}{\eta} \exp\left(\frac{-r_i^2(\xi)}{\eta}\right) d\eta \, d\xi \tag{4.36d}$$

$$I_{5,i}^{tr} = \int_0^1 \int_0^1 \frac{1}{\eta^2} \exp\left(\frac{-r_i^2(\xi)}{\eta}\right) d\eta \, d\xi \tag{4.36e}$$

$$I_{6,i}^{tr} = \int_0^1 \int_0^1 \frac{\xi}{\eta^2} \exp\left(\frac{-r_i^2(\xi)}{\eta}\right) d\eta \, d\xi \tag{4.36f}$$

where

$$r_i^2(\xi) = \frac{r^2(\xi)}{4\alpha(t_m - t_i)} \tag{4.37}$$

Furthermore, the time integration is performed analytically to yield

$$I_{1,i}^{tr} = \int_0^1 \mathrm{E}_2\left(r_i^2(\xi)\right) d\xi \tag{4.38a}$$

$$I_{2,i}^{tr} = \int_0^1 \xi\, \mathrm{E}_2\left(r_i^2(\xi)\right) d\xi \tag{4.38b}$$

$$I_{3,i}^{tr} = \int_0^1 \mathrm{E}_1\left(r_i^2(\xi)\right) d\xi \tag{4.38c}$$

$$I_{4,i}^{tr} = \int_0^1 \xi\, \mathrm{E}_1\left(r_i^2(\xi)\right) d\xi \tag{4.38d}$$

$$I_{5,i}^{tr} = \int_0^1 \frac{1}{r_i^2(\xi)} \exp\left(r_i^2(\xi)\right) d\xi \tag{4.38e}$$

$$I_{6,i}^{tr} = \int_0^1 \frac{\xi}{r_i^2(\xi)} \exp\left(r_i^2(\xi)\right) d\xi \tag{4.38f}$$

where E_1 and E_2 are exponential-integral functions whose expressions are given in reference 12. The integration over ξ is performed numerically using Gaussian quadrature with the distance function given by equation (3.21).

Summarizing up to this point, interpolation functions that are linear in both time and space are given for the dependent variables. The interpolation functions are expressed in terms of the transform coordinates ξ and η. The multiplication of the interpolation functions to the kernels of the integral transform (i.e., the weight function and its derivative) yield three core temporal integrals as functions of η. The limits of integration and the time difference in the exponential term of the core integrals are standardized. These three core time integrals are combined with the spatial integration to produce six component integrals in space and time. The temporal integration is performed analytically and the spatial integration is computed using Gaussian quadrature. Utilizing the definitions of the six integrals, the expression for the integral transforms of equations (4.29) are

$$\int_e \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} W^{tr}(r, t_m - t) \frac{\partial T_e^i}{\partial n} d\tau \, d\Gamma$$

$$= \frac{l_e}{4\pi(1-b-a)} \sum_{i=1}^{m} \frac{1}{t_i - t_{i-1}} \Bigg[\Bigg\{ -(t_m - t_i) \left[(1-b) \frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) - a \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1}) \right]$$

$$+ (t_m - t_{i-1}) \left[(1-b) \frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) - a \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i) \right] \Bigg\} \left(I_{3,i-1}^{tr} - I_{3,i}^{tr} \right)$$

$$+ \left[(1-b) \frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) - a \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1}) - (1-b) \frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) + a \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i) \right] \left[(t_m - t_{i-1}) I_{1,i-1}^{tr} - (t_m - t_i) I_{1,i}^{tr} \right]$$

$$+ \left\{ (t_m - t_i) \left[\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) - \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1}) \right] - (t_m - t_{i-1}) \left[\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) - \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i) \right] \right\} \left(I_{4,i-1}^{tr} - I_{4,i}^{tr} \right)$$

$$+ \left[-\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) + \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1}) + \frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) - a \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i) \right] \left[(t_m - t_{i-1}) I_{2,i-1}^{tr} - (t_m - t_i) I_{2,i}^{tr} \right] \Bigg] \quad (4.39a)$$

$$\int_e \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} T_e^i \frac{\partial W^{tr}(r, t_m - t)}{\partial n} d\tau \, d\Gamma$$

$$= \frac{-(\vec{r} \bullet n) l_e}{8\pi\alpha(1-b-a)} \sum_{i=1}^{m} \frac{1}{t_i - t_{i-1}} \Bigg[\Bigg\{ -(t_m - t_i) \left[(1-b) \tilde{T}_{e,1}(t_{i-1}) - a \tilde{T}_{e,2}(t_{i-1}) \right]$$

$$+ (t_m - t_{i-1}) \left[(1-b) \tilde{T}_{e,1}(t_i) - a \tilde{T}_{e,2}(t_i) \right] \Bigg\} \left[\frac{I_{5,i-1}^{tr}}{(t_m - t_{i-1})} - \frac{I_{5,i}^{tr}}{(t_m - t_i)} \right]$$

$$+ \left[(1-b) \tilde{T}_{e,1}(t_{i-1}) - a \tilde{T}_{e,2}(t_{i-1}) - (1-b) \tilde{T}_{e,1}(t_i) + a \tilde{T}_{e,2}(t_i) \right] \left(I_{3,i-1}^{tr} - I_{3,i}^{tr} \right)$$

$$+ \left\{ (t_m - t_i) \left[\tilde{T}_{e,1}(t_{i-1}) - \tilde{T}_{e,2}(t_{i-1}) \right] - (t_m - t_{i-1}) \left[\tilde{T}_{e,1}(t_i) - \tilde{T}_{e,2}(t_i) \right] \right\} \left[\frac{I_{6,i-1}^{tr}}{(t_m - t_{i-1})} - \frac{I_{6,i}^{tr}}{(t_m - t_i)} \right]$$

$$+ \left[-\tilde{T}_{e,1}(t_{i-1}) + \tilde{T}_{e,2}(t_{i-1}) + \tilde{T}_{e,1}(t_i) - a \tilde{T}_{e,2}(t_i) \right] \left(I_{4,i-1}^{tr} - I_{4,i}^{tr} \right) \Bigg] \quad (4.39b)$$

Notice that the integrals $I_{j,i}^{tr}$ appear in pairs of the form $(c_1 I_{j,i-1}^{tr} - c_2 I_{j,i}^{tr})$ because of the splitting of the integration limits shown in equation (4.34). Also, terms of the form $(t_m - t_i)$ result from the standardization given by equation (4.35). The equations are tedious and the notation is abstruse, yet the concept is straightforward and should not be lost in the details of the algebra. The equations are obtained simply by multiplying the interpolation function by the transform kernels.

Equations (4.39) are valid for any distribution of the time steps. Recall, though, that if values of t_m are selected arbitrarily, then the integrals $I_{j,i}^{tr}$ must be calculated for all possible combinations of $(t_m - t_i)$ for $0 < i < m$ and $1 \le m \le m_{\max}$. If the time step is constant, however, then $(t_i - t_{i-1})$ becomes Δt and $(t_m - t_i)$ becomes an integer multiple of Δt. The integrals need to be calculated only $m_{\max}$ times. Furthermore, because the influence coefficients are a function of a constant Δt, often only one matrix inversion is required for all time steps. Extracting out Δt, the integrals over time are

$$\int_e \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} W^{tr}(r, t_m - t) \frac{\partial T_e^i}{\partial n} d\tau d\Gamma = \frac{l_e}{4\pi(1-b-a)} \sum_{i=1}^{m} \Bigg[\Bigg\{ -(m-i)\left[(1-b)\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) - a\frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1})\right]$$

$$+(m-i+1)\left[(1-b)\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) - a\frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i)\right]\Bigg\}\left(I_{3,i-1}^{tr} - I_{3,i}^{tr}\right)$$

$$+\left[(1-b)\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) - a\frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1}) - (1-b)\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) + a\frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i)\right]\left[(m-i+1)I_{1,i-1}^{tr} - (m-i)I_{1,i}^{tr}\right]$$

$$+\left\{(m-i)\left[\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) - \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1})\right] - (m-i+1)\left[\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) - \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i)\right]\right\}\left(I_{4,i-1}^{tr} - I_{4,i}^{tr}\right)$$

$$+\left[-\frac{\partial \tilde{T}_{e,1}}{\partial n}(t_{i-1}) + \frac{\partial \tilde{T}_{e,2}}{\partial n}(t_{i-1}) + \frac{\partial \tilde{T}_{e,1}}{\partial n}(t_i) - a\frac{\partial \tilde{T}_{e,2}}{\partial n}(t_i)\right]\left[(m-i+1)I_{2,i-1}^{tr} - (m-i)I_{2,i}^{tr}\right] \Bigg] \tag{4.40a}$$

$$\int_e \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} T_e^i \frac{\partial W^{tr}(r, t_m - t)}{\partial n} d\tau d\Gamma = \frac{-(\vec{r} \bullet n) l_e}{8\pi\alpha\Delta t(1-b-a)} \sum_{i=1}^{m} \Bigg[\Big\{ -(m-i)\Big[(1-b)\tilde{T}_{e,1}(t_{i-1}) - a\tilde{T}_{e,2}(t_{i-1})\Big]$$

$$+(m-i+1)\Big[(1-b)\tilde{T}_{e,1}(t_i) - a\tilde{T}_{e,2}(t_i)\Big]\Big\}\left[\frac{I_{5,i-1}^{tr}}{(m-i+1)} - \frac{I_{5,i}^{tr}}{(m-i)}\right]$$

$$+\Big[(1-b)\tilde{T}_{e,1}(t_{i-1}) - a\tilde{T}_{e,2}(t_{i-1}) - (1-b)\tilde{T}_{e,1}(t_i) + a\tilde{T}_{e,2}(t_i)\Big]\Big(I_{3,i-1}^{tr} - I_{3,i}^{tr}\Big)$$

$$+\Big\{(m-i)\Big[\tilde{T}_{e,1}(t_{i-1}) - \tilde{T}_{e,2}(t_{i-1})\Big] - (m-i+1)\Big[\tilde{T}_{e,1}(t_i) - \tilde{T}_{e,2}(t_i)\Big]\Big\}\left[\frac{I_{6,i-1}^{tr}}{(m-i+1)} - \frac{I_{6,i}^{tr}}{(m-i)}\right]$$

$$+\Big[-\tilde{T}_{e,1}(t_{i-1}) + \tilde{T}_{e,2}(t_{i-1}) + \tilde{T}_{e,1}(t_i) - a\tilde{T}_{e,2}(t_i)\Big]\Big(I_{4,i-1}^{tr} - I_{4,i}^{tr}\Big) \Bigg] \tag{4.40b}$$

4.4.2 Singular Integrals– The previous section develops expressions for the integral transforms when the integrand is nonsingular; this section presents expressions for the integral transforms for a singular integrand. The singularity occurs when the source point lies on the element defining the path of integration. Because the distance function simplifies in this orientation, the integral transform admits a complete analytic solution over time and space. Furthermore, since the source point lies along the path of integration ($\vec{r} \bullet \hat{n} = 0$), the integral of equation (4.29a) is zero.

The singular integral requires an interpolation function for $\partial\tilde{T}_e^i / \partial n$. It is convenient to redefine the spatial transformation such that the origin lies on the source node point. The range of the transformation becomes $-a \le \xi \le 1-a$, where a is the offset distance of the source point. The notation in this section is analogous to the notation used for *Orientation 3* of the steady-state transform of Section 3.5.3. Letting subscript p denote the source point and q denote the other node point, the interpolation function for $\partial\tilde{T}_e^i / \partial n$ is

$$\frac{\partial\tilde{T}_e^i}{\partial n} = \frac{1}{(1-b-a)} \Bigg\{ \chi_i \left[(1-b-a-\xi) \frac{\tilde{T}_{e,p}}{\partial n}(t_{i-1}) + \xi \frac{\tilde{T}_{e,q}}{\partial n}(t_{i-1}) \right] + (1-\chi_i) \left[(1-b-a-\xi) \frac{\tilde{T}_{e,p}}{\partial n}(t_i) + \xi \frac{\tilde{T}_{e,q}}{\partial n}(t_i) \right] \Bigg\} \tag{4.41}$$

In the new transformation, the second node point is located at $\xi = 1-a-b$.

Similar to the derivation in the previous section, multiplication of the interpolation function to the transformed fundamental solution produces component integrals over time and space like the integrals of equations (4.36). With the revised spatial transformation, the component integrals are

$$I_{1,i}^{tr} = \int_{-a}^{1-a} \int_{0}^{1} \exp\left(\frac{-r_i^2(\xi)}{\eta} \right) d\eta \, d\xi \tag{4.42a}$$

$$I_{2,i}^{tr} = \int_{-a}^{1-a} \int_{0}^{1} \xi \exp\left(\frac{-r_i^2(\xi)}{\eta} \right) d\eta \, d\xi \tag{4.42b}$$

$$I_{3,i}^{tr} = \int_{-a}^{1-a} \int_{0}^{1} \frac{1}{\eta} \exp\left(\frac{-r_i^2(\xi)}{\eta} \right) d\eta \, d\xi \tag{4.42c}$$

$$I_{4,i}^{tr} = \int_{-a}^{1-a} \int_{0}^{1} \frac{\xi}{\eta} \exp\left(\frac{-r_i^2(\xi)}{\eta} \right) d\eta \, d\xi \tag{4.42d}$$

Because only one of the integral transforms is nonzero, only four component integrals are needed (as opposed to the six component integrals for the nonsingular case given by equations (4.36)). Each of the four integrals admits an analytic solution over time and space. The derivation of $I_{1,i}^{tr}$ is presented in detail as an example. The other three are derived similarly, but, for brevity, only the final expressions are given. With the distance function explicitly expressed in terms of ξ, $I_{1,i}^{tr}$ is given by

$$I_{1,i}^{tr} = \int_{0}^{1} \int_{-a}^{1-a} \exp\left(\frac{-\beta_i \xi^2}{\eta} \right) d\xi \, d\eta \tag{4.43}$$

where

$$\beta_i = \frac{l_e}{4\alpha(t_m - t_i)} \tag{4.44}$$

The integral is singular at the origin. An additional transformation to polar coordinates allows the Cauchy principle value to be computed in the limit as the radius goes to zero. The domain is divided into two regions along the η axis. A separate polar coordinate transformation is prescribed for each region, as shown in figure 8. In each region, the limits of integration are easily determined by splitting the integral in two at the corners of ξ-η space; i.e., at $(\xi, \eta) = (1-a, 1)$ and at $(\xi, \eta) = (-a, 1)$. Both regions are transformed to polar coordinates and subdivided in two to yield four separate integrals

$$
\begin{aligned}
I_{1,i}^{tr} = & \int_0^{\tan^{-1}\left(\frac{1}{1-a}\right)} \int_0^{\frac{1-a}{\cos\theta}} \exp\left(\frac{-\beta_i \lambda \cos^2\theta}{\sin\theta}\right) \lambda \, d\lambda \, d\theta \\
& + \int_{\tan^{-1}\left(\frac{1}{1-a}\right)}^{\frac{\pi}{2}} \int_0^{\frac{1}{\sin\theta}} \exp\left(\frac{-\beta_i \lambda \cos^2\theta}{\sin\theta}\right) \lambda \, d\lambda \, d\theta \\
& + \int_0^{\tan^{-1}(a)} \int_0^{\frac{1}{\cos\psi}} \exp\left(\frac{-\beta_i \lambda \sin^2\psi}{\cos\psi}\right) \lambda \, d\lambda \, d\psi \\
& + \int_{\tan^{-1}(a)}^{\frac{\pi}{2}} \int_0^{\frac{a}{\sin\psi}} \exp\left(\frac{-\beta_i \lambda \sin^2\psi}{\cos\psi}\right) \lambda \, d\lambda \, d\psi
\end{aligned}
\tag{4.45}
$$

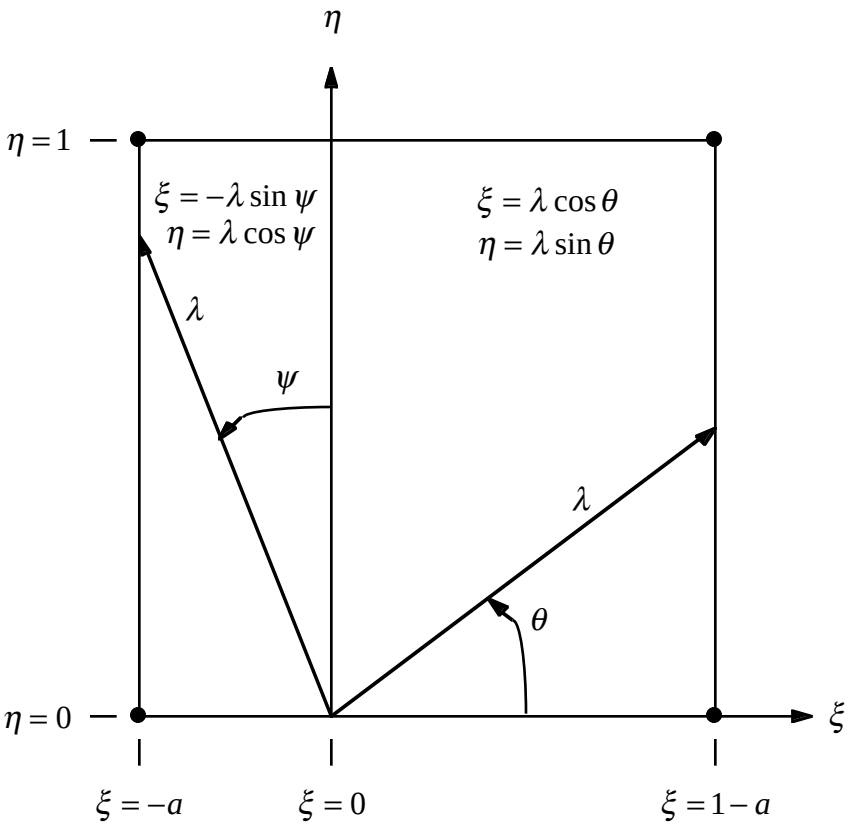

Figure 8. Singular integration domain and polar transformation.

The inner integration over λ is performed analytically to yield

$$
\begin{aligned}
I_{1,i}^{tr} = &\int_0^{\tan^{-1}\left(\frac{1}{1-a}\right)} \frac{-\sin^2\theta}{\beta_i^2\cos^4\theta}\left[\exp\left(\frac{-\beta_i(1-a)\cos\theta}{\sin\theta}\right)\left(\frac{\beta_i(1-a)\cos\theta}{\sin\theta}+1\right)-1\right]d\theta \\
&+\int_{\tan^{-1}\left(\frac{1}{1-a}\right)}^{\frac{\pi}{2}} \frac{-\sin^2\theta}{\beta_i^2\cos^4\theta}\left[\exp\left(\frac{-\beta_i\cos^2\theta}{\sin^2\theta}\right)\left(\frac{\beta_i\cos^2\theta}{\sin^2\theta}+1\right)-1\right]d\theta \\
&+\int_0^{\tan^{-1}(a)} \frac{-\cos^2\psi}{\beta_i^2\sin^4\psi}\left[\exp\left(\frac{-\beta_i\sin^2\psi}{\cos^2\psi}\right)\left(\frac{\beta_i\sin^2\psi}{\cos^2\psi}+1\right)-1\right]d\psi \\
&+\int_{\tan^{-1}(a)}^{\frac{\pi}{2}} \frac{-\cos^2\psi}{\beta_i^2\sin^4\psi}\left[\exp\left(\frac{-\beta_i a\sin\psi}{\cos\psi}\right)\left(\frac{\beta_i a\sin\psi}{\cos\psi}+1\right)-1\right]d\psi
\end{aligned}
\tag{4.46}
$$

Each of the four integrals of the right-hand side of equation (4.46) is mapped to a dummy variable, *u*, through separate substitutions. Letting $u=\tan\theta/(1-a)$, $u=\tan\theta$, $u=\cot\psi$, and $u=\cot\psi/a$ for the first through fourth integral terms, respectively, yields

$$
\begin{aligned}
I_{1,i}^{tr} = &-(1-a)^3\int_0^{\frac{1}{(1-a)^2}} \frac{u^2}{\beta_i^2}\left[\exp\left(\frac{-\beta_i}{u}\right)\left(\frac{\beta_i}{u}+1\right)-1\right]du \\
&-\int_{\frac{1}{1-a}}^{\infty} \frac{u^2}{\beta_i^2}\left[\exp\left(\frac{-\beta_i}{u^2}\right)\left(\frac{\beta_i}{u^2}+1\right)-1\right]du \\
&-\int_{\frac{1}{a}}^{\infty} \frac{u^2}{\beta_i^2}\left[\exp\left(\frac{-\beta_i}{u^2}\right)\left(\frac{\beta_i}{u^2}+1\right)-1\right]du \\
&-a^3\int_0^{\frac{1}{a^2}} \frac{u^2}{\beta_i^2}\left[\exp\left(\frac{-\beta_i}{u}\right)\left(\frac{\beta_i}{u}+1\right)-1\right]du
\end{aligned}
\tag{4.47}
$$

The first and fourth integrals on the right-hand side of equation (4.47) are of the same form, and the second and third integrals are of the same form. The solutions of the third and fourth integrals are addressed with the knowledge that the solutions of the first and second are obtained by substitution of (1 – *a*) for *a*. The fourth integral of equation (4.47) yields to analytic solution. Furthermore, the solution is expanded into series form; the result is

$$-a^3\int_0^{\frac{1}{a^2}}\frac{u^2}{\beta_i^2}\left[\exp\left(\frac{-\beta_i}{u}\right)\left(\frac{\beta_i}{u}+1\right)-1\right]du=\frac{1}{3}\left[\frac{1}{a^3\beta_i^2}-\frac{1}{a\beta_i}\exp\left(-a^2\beta_i\right)\left(1+\frac{1}{a^2\beta_i}\right)+a\,\mathrm{E}_2(a^2\beta_i)\right]$$

$$=\frac{1}{3}\left[\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k a^{2k+1}}{(k+2)k!}+a\,\mathrm{E}_2(a^2\beta_i)\right] \tag{4.48}$$

The third integral does not readily submit to analytic solution. Nevertheless, the integrand is expanded into series form and each term is integrable. The Cauchy principle value is found for each term at the upper limit of integration; the result is

$$-\int_{\frac{1}{a}}^{\infty}\frac{u^2}{\beta_i^2}\left[\exp\left(\frac{-\beta_i}{u^2}\right)\left(\frac{\beta_i}{u^2}+1\right)-1\right]du=\operatorname*{Lim}_{\gamma\to\infty}\int_{\frac{1}{a}}^{\gamma}\sum_{k=1}^{\infty}\frac{(-1)^{k-1}\beta_i^{k-1}}{u^{2k}(k+1)(k-1)!}\,du$$

$$=\operatorname*{Lim}_{\gamma\to\infty}\sum_{k=0}^{\infty}\frac{(-1)^{k+1}\beta_i^k}{u^{2k+1}(2k+1)(k+2)k!}\Bigg|_{\frac{1}{a}}^{\gamma} \tag{4.49}$$

$$=\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k a^{2k+1}}{(2k+1)(k+2)k!}$$

The two series expansions found in equations (4.48) and (4.49) are combined to yield

$$\frac{1}{3}\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k a^{2k+1}}{(k+2)k!}+\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k a^{2k+1}}{(2k+1)(k+2)k!}=\frac{2}{3}\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k a^{2k+1}}{(2k+1)k!} \tag{4.50}$$

Since the two series solutions can be combined, the sum of the solutions of the third and fourth integrals renders two terms: a series expansion given by equation (4.50) and an exponential integral function. The first and second integrals combine in like fashion. Summing all four integrals, the solution for $I_{1,i}^{tr}$ is

$$I_{1,i}^{tr}=\frac{1-a}{3}\mathrm{E}_2\left((1-a)^2\beta_i\right)+\frac{a}{3}\mathrm{E}_2(a^2\beta_i)+\frac{2}{3}\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k\left[(1-a)^{2k+1}+a^{2k+1}\right]}{(2k+1)k!} \tag{4.51a}$$

Similar derivations for $I_{2,i}^{tr}$, $I_{3,i}^{tr}$, and $I_{4,i}^{tr}$ yield

$$I_{2,i}^{tr}=\frac{(1-a)^2}{4}\mathrm{E}_2\left((1-a)^2\beta_i\right)-\frac{a^2}{4}\mathrm{E}_2(a^2\beta_i)+\frac{1}{4}\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k\left[(1-a)^{2k+2}-a^{2k+2}\right]}{(k+1)!} \tag{4.51b}$$

$$I_{3,i}^{tr} = \frac{1}{\beta_i}\left[\frac{1-\mathrm{E}_2\left((1-a)^2\beta_i\right)}{1-a} + \frac{1-\mathrm{E}_2\left(a^2\beta_i\right)}{a}\right] + \sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k\left[(1-a)^{2k+1}+a^{2k+1}\right]}{(2k+1)(k+1)!} \tag{4.51c}$$

$$I_{4,i}^{tr} = \frac{1}{2\beta_i}\left[\left(1-\mathrm{E}_2\left((1-a)^2\beta_i\right)\right) - \left(1-\mathrm{E}_2\left((1-a)^2\beta_i\right)\right)\right] \tag{4.51d}$$

The series shown in equations (4.51) are convergent for $\beta_i < 1$. In practice, the element is broken into subelements such that the length of the subelement containing the singularity yields $\beta_i = 0.8$. For $\beta_i = 0.8$, the series converges to sufficient accuracy in six terms. The remaining subelements are nonsingular and are integrated numerically as in the previous section.

When $a = 0$, the expressions of equations (4.51) are undefined; however, the limiting form is found to be

$$I_{1,i}^{tr} = \mathrm{E}_2(\beta_i) + \frac{2}{3}\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k}{(2k+1)k!} \tag{4.52a}$$

$$I_{2,i}^{tr} = \frac{1}{4}\mathrm{E}_2(\beta_i) + \frac{1}{4}\sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k}{(k+1)!} \tag{4.52b}$$

$$I_{3,i}^{tr} = \frac{1-\mathrm{E}_2(\beta_i)}{\beta_i} + \sum_{k=0}^{\infty}\frac{(-1)^k\beta_i^k}{(2k+1)(k+1)!} \tag{4.52c}$$

$$I_{4,i}^{tr} = \frac{\left(1-\mathrm{E}_2(\beta_i)\right)}{2\beta_i} \tag{4.52d}$$

The analytic expressions given by equations (4.51) and (4.52) are compared to numerical integration in figures 9 through 12. The numerical integration is performed using two-dimensional Simpson's Rule with 10^9 functional evaluations. The analytic expressions are seen to reproduce the numerical values over a range of β_i and a. Recall that the analytic expressions are not employed for $\beta_i > 0.8$ to ensure convergence of the series solution. The only discrepancy between the analytical and numerical solutions is for $I_{3,i}^{tr}$ at small values of β_i, as shown in figure 11. This difference is caused by error in the numerical solution; the Simpson integration requires extremely fine spacing to resolve the singularity of the integrand.

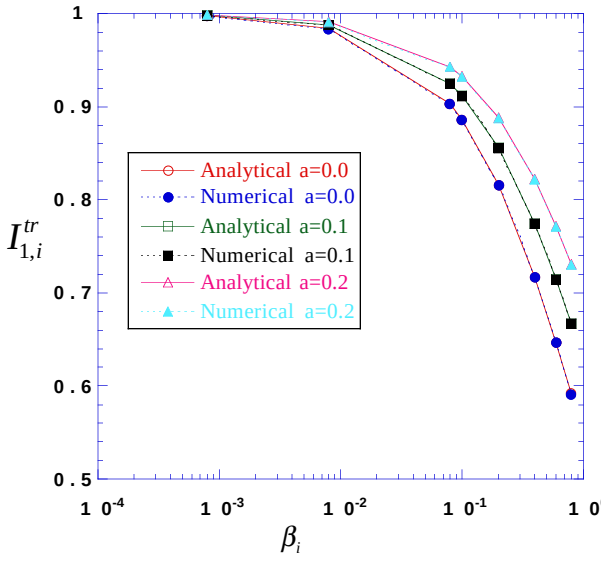

Figure 9. Comparison of analytical to numerical solution of transient, singular, component integral $I^{tr}_{1,i}$.

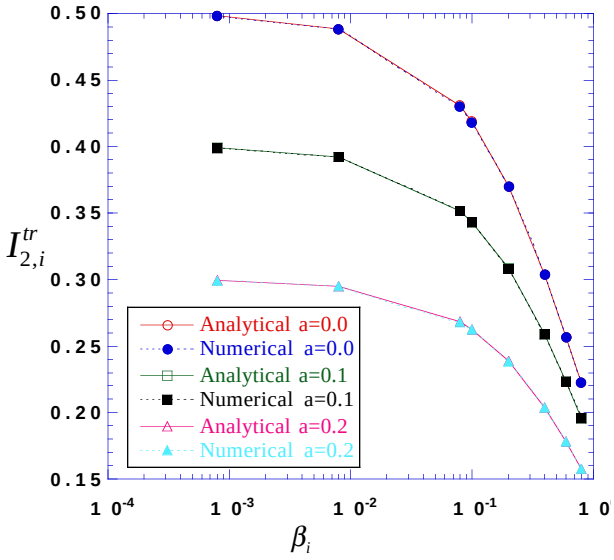

Figure 10. Comparison of analytical to numerical solution of transient, singular, component integral $I^{tr}_{2,i}$.

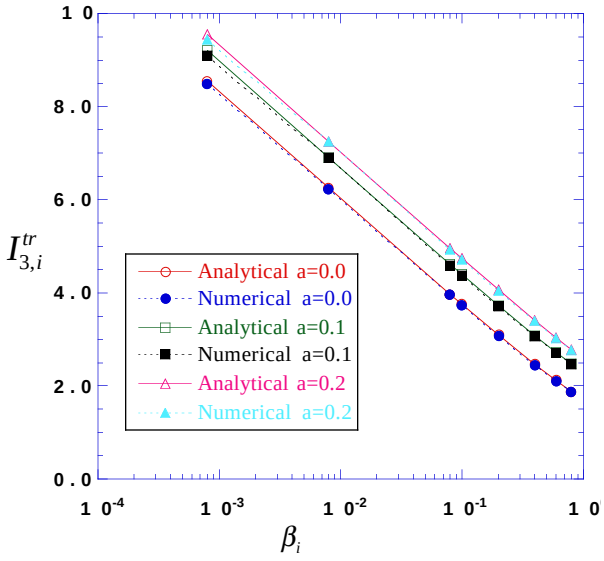

Figure 11. Comparison of analytical to numerical solution of transient, singular, component integral $I_{3,i}^{tr}$.

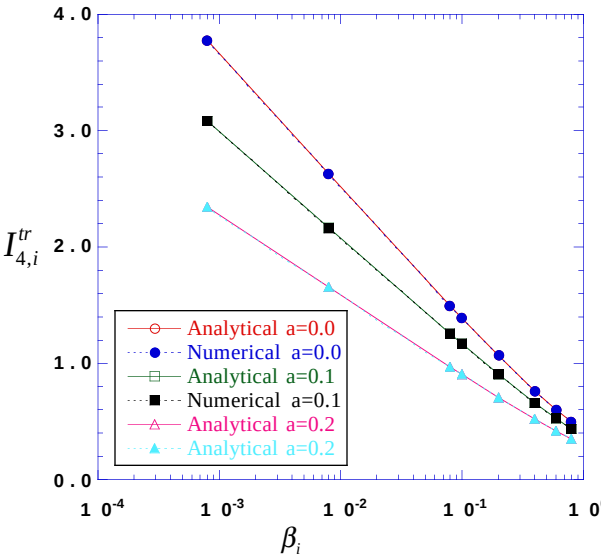

Figure 12. Comparison of analytical to numerical solution of transient, singular, component integral $I_{4,i}^{tr}$.

With $I^{tr}_{j,i}$ defined, the expression for the singular integral transform is

$$\int_e \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} W^{tr}(r, t_m - t) \frac{\partial T^i_e}{\partial n} d\tau d\Gamma$$

$$= \frac{l_e}{4\pi(1-b-a)} \sum_{i=1}^{m} \frac{1}{t_i - t_{i-1}} \Bigg[(1-b-a) \left[-(t_m - t_i) \frac{\partial \tilde{T}_{e,p}}{\partial n}(t_{i-1}) + (t_m - t_{i-1}) \frac{\partial \tilde{T}_{e,p}}{\partial n}(t_i) \right] \left(I^{tr}_{3,i-1} - I^{tr}_{3,i} \right)$$

$$+ (1 - b - a) \left[\frac{\partial \tilde{T}_{e,p}}{\partial n}(t_{i-1}) - \frac{\partial \tilde{T}_{e,p}}{\partial n}(t_i) \right] \left[(t_m - t_{i-1}) I^{tr}_{1,i-1} - (t_m - t_i) I^{tr}_{1,i} \right]$$

$$+ \left\{ (t_m - t_i) \left[\frac{\partial \tilde{T}_{e,p}}{\partial n}(t_{i-1}) - \frac{\partial \tilde{T}_{e,q}}{\partial n}(t_{i-1}) \right] - (t_m - t_{i-1}) \left[\frac{\partial \tilde{T}_{e,p}}{\partial n}(t_i) - \frac{\partial \tilde{T}_{e,q}}{\partial n}(t_i) \right] \right\} \left(I^{tr}_{4,i-1} - I^{tr}_{4,i} \right)$$

$$+ \left[-\frac{\partial \tilde{T}_{e,p}}{\partial n}(t_{i-1}) + \frac{\partial \tilde{T}_{e,q}}{\partial n}(t_{i-1}) + \frac{\partial \tilde{T}_{e,p}}{\partial n}(t_i) - a \frac{\partial \tilde{T}_{e,q}}{\partial n}(t_i) \right] \left[(t_m - t_{i-1}) I^{tr}_{2,i-1} - (t_m - t_i) I^{tr}_{2,i} \right] \Bigg] \tag{4.53}$$

4.5 Numerical Results

The transient algorithm previously developed is tested against three analytic solutions. All three test cases are simple one- and two-dimensional geometries for which transient analytic solutions are found in reference 13. The first case is the temperature decay in a one-dimensional rod with an initial linear temperature distribution. The second case is the temperature decay of a two-dimensional plate initially at uniform temperature. Finally, the third case is a one-dimensional rod with a time-dependent fluctuating temperature boundary condition.

4.5.1 One-Dimensional Rod– The first test case is a one-dimensional rod of length *l* with an initial linear temperature profile. The initial and boundary conditions are given by

Initial conditions: $T(x, t = 0) = (l - x)/l$ for $0 \le x \le l$

Boundary conditions: $\partial T / \partial n (x = 0, t) = 0$ for $t > 0$

$T(x = l, t) = 0$ for $t > 0$

The domain is discretized using 11 elements along the length of the rod and 5 elements of length $0.1l$ in the second dimension. Symmetry boundary conditions are imposed along $y = 0$ and $y = 0.5l$ to simulate a one-dimensional problem. Temperature values at five select points along the length of the domain are compared to the analytic solution in figure 13. The abscissa is time nondimensionalized by the thermal diffusivity, α, and the length of the rod. The boundary element solution reproduces

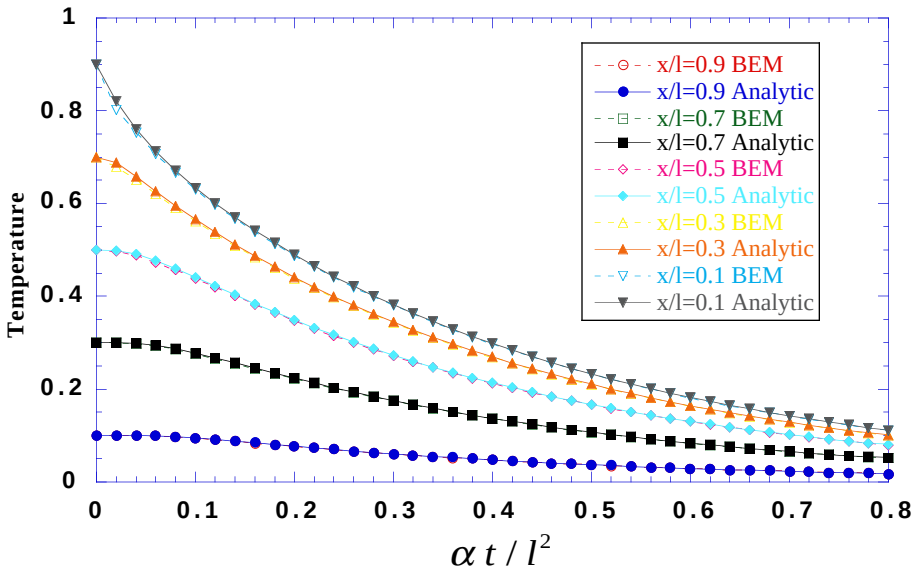

Figure 13. Comparison of analytical to boundary element transient temperature profiles in one-dimensional rod.

the analytic solution throughout the domain and for all time. The error, expressed as a percentage of the difference between the analytical and numerical solutions, is plotted in figure 14. The greatest error occurs at the initial time step; the cause of this effect is discussed in the next section. After the initial time step, the error decays to within 0.3 percent of the analytical solution.

4.5.2 Two-Dimensional Plate– The second test case is a finite square with unit initial temperature. At time zero, the boundary temperature is instantaneously dropped to zero. Heat then flows out of the domain and the temperature drops to zero in the steady state. For this simulation, the boundary is discretized using 11 equally spaced elements per side. The square ranges from $(x,y)=(0,0)$ to $(x,y)=(l,l)$. Symmetry boundary conditions are imposed along $x=0$ and $y=0$. The temperature is set to zero on the remaining two sides. In figure 15, the boundary element solution is compared to the analytic solution at five locations inside the domain. The points that are plotted, although selected for broad representation of the solution, are completely arbitrary and have no correspondence to any type of interior discretization. Any other five points could easily have been selected. Except for the first two or three time steps, the boundary element and analytic solutions are nearly identical to plotting accuracy. The error of the solution is presented in figure 16. Similar to the results presented in figure 14, the greatest error occurs at the initial time step. The error is slowly rising in time past 0.2 because the error is expressed as a percentage of the analytical solution, which is approaching zero. Thus, the percentage error is increasing even though the absolute error is decreasing.

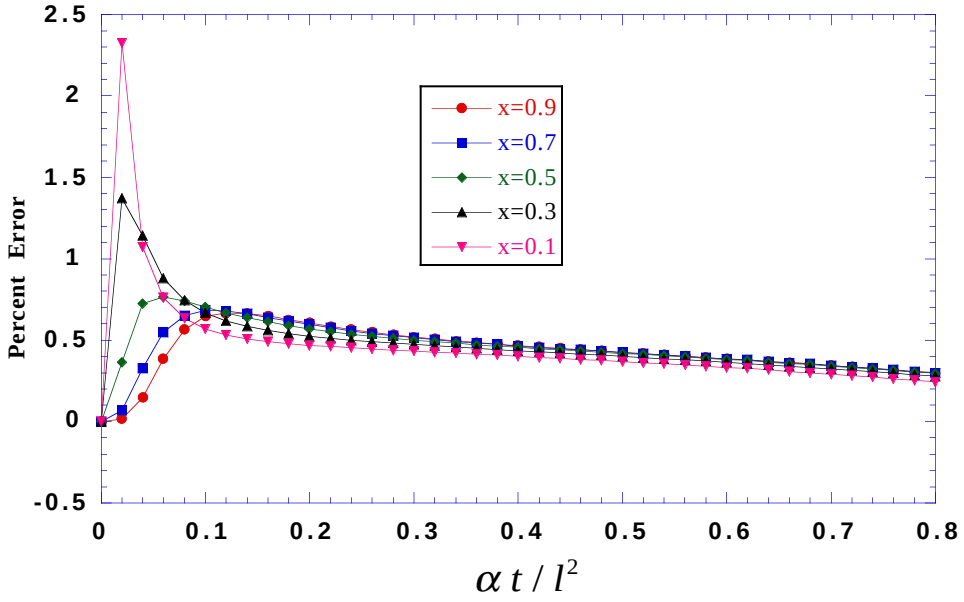

Figure 14. Error of boundary element temperature profiles in a one-dimensional rod.

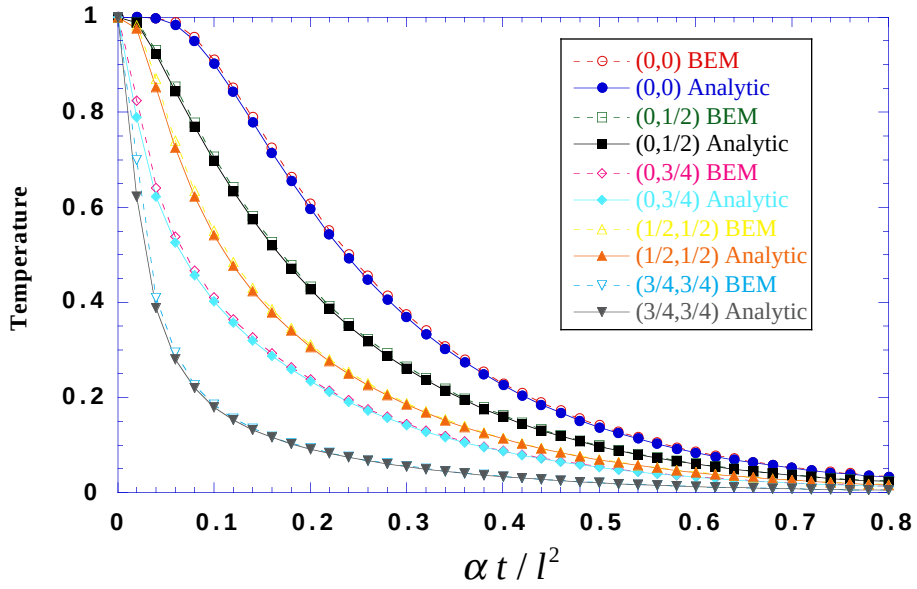

Figure 15. Comparison of analytical to boundary element transient temperature profiles in a two-dimensional plate.

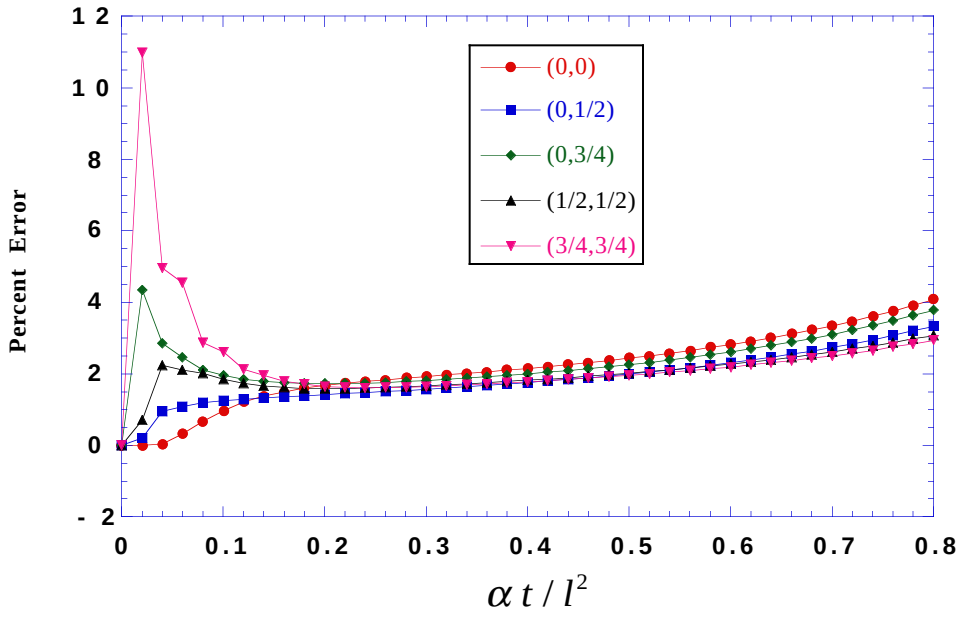

Figure 16. Error of boundary element temperature profiles in a two-dimensional plate.

The slight deviations near time zero are explained by recognizing that the heat flux at time zero is infinite because of the instantaneous temperature drop. Consequently, the initial temperature gradient is computationally undefined which undermines the ability of the linear shape function to describe the thermal gradient over the first time step. This computational obstacle is circumvented by dropping the order of the temporal shape function over the first time step. Thus, the first time step for the transient calculation employs a constant temporal shape function while all subsequent time steps use a linear. As a result, the first time step has a reduction in accuracy as compared to the remainder of the computation. The effects of the accuracy reduction, however, do not propagate in time.

4.5.3 One-Dimensional Rod with Fluctuating Boundary Condition– The final test case is a one-dimensional slab of length l with a time-dependent, fluctuating boundary condition. The initial and boundary conditions are given by,

$$\text{Initial conditions: } T(x,t=0)=0 \text{ for } 0 \le x \le 1$$

$$\text{Boundary conditions: } T(x=0,t)=0 \text{ for } t>0$$

$$T(x=l,t)=\sin(10\pi t+\pi/2) \text{ for } t>0$$

The domain is discretized using 10 equally spaced elements on each side along the length of the domain plus 4 additional elements in the second dimension. The boundary element solution is compared to the analytic solution at 3 locations along the length of the domain in figure 17. Once again, the boundary element solution is identical to the analytic solution within plotting accuracy.

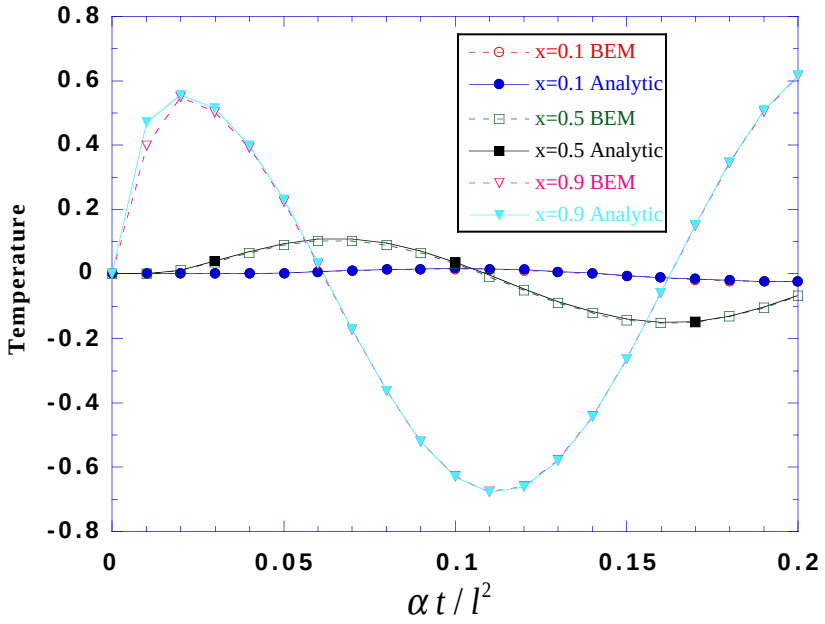

Figure 17. Comparison of analytical to boundary element transient temperature profiles in one-dimensional rod with fluctuating temperature boundary conditions.

The only marked error occurs at the first time step because of the constant temporal discretization mentioned previously. The L_2 error over all space is shown in figure 18 as a function of time and the time step. The error fluctuates at the same frequency as the boundary condition forcing function. This result is consistent with finite difference analysis, which shows the truncation error to posses the same growth property as the exact solution. (See ref. 14.) In this case, the temperature field is continually perturbed by the boundary condition, and, therefore, the solution and the error do not decay in time. Furthermore, the error decreases as the time step is decreased. The values of the peaks of the fluctuating errors are plotted as functions of the time step in figure 19. The slope of the curve shows the algorithm to be second-order accurate in time. Moreover, since the same interpolation function as the steady-state algorithm is employed, the algorithm retains second-order spatial accuracy.

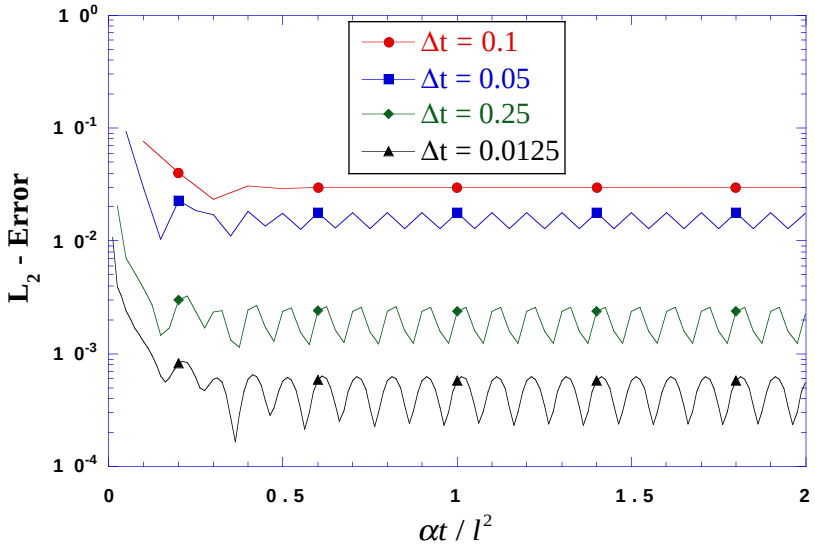

Figure 18. Error of boundary element transient temperature profiles in one-dimensional rod with fluctuating temperature boundary conditions.

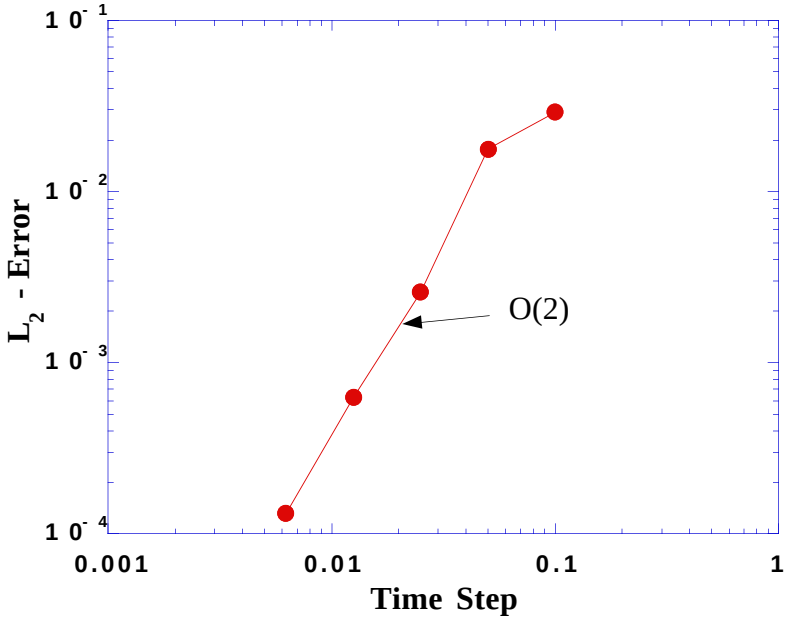

Figure 19. Error of boundary element transient temperature profiles in one-dimensional rod with fluctuating temperature boundary conditions.

5 Conclusion

Two-dimensional, steady-state, and transient boundary element algorithms are designed for coupling with CFD flow solvers. The algorithms feature linear boundary elements for efficient coupling with discretized fluid domains that typically assume linear segments between grid points. Furthermore, the solution variables of T and $\partial T/\partial n$ are approximated using a linear interpolation function with offset node points; the offset nodes are implemented to uniquely define the nodal, element, surface normal.

Analytic expressions for the requisite boundary integral transforms are derived for the steady-state algorithm. The analytic solutions are possible for both singular and nonsingular integrands because of the simplifications that result from using isoparametric linear shape functions. The steady-state algorithm is used to compute a variety of test cases that admit analytic solutions. The boundary element algorithm reproduces the analytic temperature to second-order spatial accuracy.

The transient algorithm incorporates the transient fundamental solution approach. The boundary-only character of the solution procedure is maintained by originating the integral transforms from the initial time. The dependent variable interpolation functions are linear in both space and time. A series solution of the singular integral transform is derived and verified by comparison to numerical integration. The remaining integral transforms are computed using Gaussian quadrature. The transient algorithm is shown to be second-order accurate in the computation of test cases for which analytic solutions exist.

The verification tests presented in this publication indicate that the BEM algorithms developed herein are suitable for accurate modeling of linear heat conduction. The algorithms are tailored for efficient and conservative coupling to CFD discretizations through the use of linear elements. Further modifications to the codes may be needed to incorporate boundary conditions required for coupling to a CFD code.

References

1. Löhner, R.; Yang, C.; Cebral, J.; Baum, J. D.; Luo, H; Pelessone, D.; and Charman, C.: Fluid-Structure Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids. AIAA Paper 95-2259, June 1995.

2. Li, H.; and Kassab, A. J.: A Coupled FVM/BEM Approach to Conjugate Heat Transfer in Turbine Blades. AIAA Paper 94-1981, June 1994.

3. Li, H.; and Kassab, A. J.: Numerical Prediction of Fluid Flow and Heat Transfer in Turbine Blades with Internal Cooling. AIAA Paper 94-2933, June 1994.

4. Brebbia, C. A.; Telles, J. C. F.; and Wrobel, L. C.: Boundary Element Techniques—Theory and Applications in Engineering. Springer-Verlag, Berlin-Heidelberg, 1984.

5. Becker, A. A.: The Boundary Element Method in Engineering. McGraw-Hill, New York, 1992.

6. Gipson, G. S.: Boundary Element Fundamentals—Basic Concepts and Recent Developments in the Poisson Equation. Topics in Engineering, C. A. Brebbia and J. J. Conner, eds., vol. 2, Computational Mechanics Publications, 1987.

7. Rizzo, F. J.: The Boundary Element Method Some Early History—A Personal View. In Boundary Element Methods in Structural Analysis, D. E. Beskos, ed., American Society of Civil Engineers, New York, 1991, pp. 1–16.

8. Sokolnikoff, I. S.; and Redheffer, R. M.: Mathematics of Physics and Modern Engineering. McGraw-Hill, New York, 1958.

9. Kassab, A. J.; and Nordlund, R. S.: Addressing the Corner Problem in BEM Solution of Heat Conduction Problems. Communications in Numerical Methods in Engineering, vol. 10, 1994, pp. 385–392.

10. Wrobel, L. C.; and Brebbia, C. A.: Boundary Elements in Thermal Problems. In Numerical Methods in Heat Transfer, R. W. Lewis, K. Morgan, and O. C. Zienkiewicz, eds., Wiley, Chichester, 1981, pp. 91–113.

11. Wrobel, L. C.; and Brebbia, C. A.: Time Dependent Potential Problems. In Progress in Boundary Element Methods, vol. 1, C. A. Brebbia, ed., Wiley, New York, 1981, pp. 192–212.

12. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. M. Abramowitz and I. A. Stegun, eds., Dover, New York, 1964.

13. Carslaw, H. S.; and Jaeger, J. C.: Conduction of Heat in Solids. Clarendon Press/Oxford, 2nd ed., 1959.

14. Anderson, D. A.; Tannehill, J. C.; and Pletcher, R. H.: Computational Fluid Mechanics and Heat Transfer. Hemisphere Publishing Corp., New York, 1984.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1997	Technical Memorandum

4. TITLE AND SUBTITLE

Steady-State and Transient Boundary Element Methods for Coupled Heat Conduction

5. FUNDING NUMBERS

242-80-01

6. AUTHOR(S)

Dean A. Kontinos*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

A-975389

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-110427

11. SUPPLEMENTARY NOTES

Point of Contact: Dean A. Kontinos, Ames Research Center, MS 230-2, Moffett Field, CA 94035-1000
(415) 604-4283
*Thermosciences Institute, Ames Research Center.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified — Unlimited
Subject Category 34

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

Boundary element algorithms for the solution of steady-state and transient heat conduction are presented. The algorithms are designed for efficient coupling with computational fluid dynamic discretizations and feature piecewise linear elements with offset nodal points. The steady-state algorithm employs the fundamental solution approach; the integration kernels are computed analytically based on linear shape functions, linear elements, and variably offset nodal points. The analytic expressions for both singular and nonsingular integrands are presented. The transient algorithm employs the transient fundamental solution; the temporal integration is performed analytically and the nonsingular spatial integration is performed numerically using Gaussian quadrature. A series solution to the integration is derived for the instance of a singular integrand. The boundary-only character of the algorithm is maintained by integrating the influence coefficients from initial time. Numerical results are compared to analytical solutions to verify the current boundary element algorithms. The steady-state and transient algorithms are numerically shown to be second-order accurate in space and time, respectively.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Computational heat transfer, Boundary element method	55
	16. PRICE CODE: A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		